UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN SISTEMAS DE COMUNICACIONES

TRABAJO FIN DE GRADO

DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE VOIP BASADO EN ASTERISK Y PBX

AUTOR: Borja García de Vinuesa Ordovás
TUTOR: Ricardo Romeral Ortega

I. INTRODUCCIÓN

Desde los principios de la humanidad la comunicación ha sido el pilar ante el cual todos los demás elementos sociales han sido construidos. Sin la comunicación, el ser humano es desprovisto de una herramienta transmisora de ideas, sentimientos e información. Su uso ha sido continuo desde la aparición del ser humano hace más de dos millones de años.

Primero fueron los signos y señales, reflejadas en las diferentes formas del arte prehistórico o el uso del fuego para la comunicación a través de señales de humo. A medida que la sociedad fue avanzando surgieron nuevos medios como el telégrafo o la base de las comunicaciones modernas, la red telefónica. Sin ella, gran porcentaje de las actividades empresariales no tendría lugar o al menos, su nivel productivo quedaría desamparado al intentar seguir el ritmo impuesto por la sociedad de nuestros días, sociedad a la que muchos se refieren de manera acertada como “sociedad de la información”.

Una vez sobrepasado el obstáculo de la distancia y con el surgimiento de la era informática, no se tardó en idear mecanismos con los cuales poder comunicar más, en menos tiempo y a mayor distancia. Surge Internet.

Gracias a esta Red de redes las comunicaciones han tomado otro cariz. Ya no se trata sólo de establecer una comunicación para dar una orden, mantener una conversación o establecer contacto de diferente índole; ahora manejamos datos. Millones de datos viajan por Internet abriendo a su paso una nueva era en la que el modelo socio-económico está cambiando y con él, todos los aspectos que lo rodean.

La voz ya no es el objeto principal de las comunicaciones. Los días en los que vivimos están inundados de datos "empaquetados" y la voz no es una excepción. Durante años las compañías telefónicas han impuesto las condiciones a la hora de tarificar nuestras conversaciones a través de sus redes. Esto no ha variado ya que de manera similar nos cobran por los datos transmitidos o recibidos por o en nuestro punto de acceso. Sin embargo, los recursos dedicados a cada cliente si han cambiado. Para la transmisión de datos (Packet Data) no es necesario disponer de circuitos dedicados; esto repercute directamente en el coste final habiendo desencadenado el desarrollo de tecnologías como la VozIP.

¿Qué es VoIP?

El término VoIP, en inglés *Voice over Internet Protocol,* está referido al conjunto de tecnologías y elementos implicados que hacen posible la comunicación de voz o de sesiones multimedia a través de redes basada en IP como es Internet.

Frente a la telefonía tradicional, en la que se emplea conmutación de circuitos, VoIP se basa en la conmutación de paquetes mediante el uso del protocolo IP. Debido al funcionamiento básico de Internet en el que no se garantiza la recepción exitosa de todos los paquetes enviados (*best effort*), si hablamos de VoIP cobran mayor importancia factores como la latencia o *jitter*[1], factores inapreciables o inexistentes al hablar de conmutación de circuitos.

[1] Variación temporal en la recepción de paquetes.

[2] Se entiende comportamiento telefónico como los hábitos de uso de la empresa. **Cómo** llaman las personas que usan el sistema telefónico.

En una Red Telefónica Conmutada son las centrales de conmutación las encargadas de establecer una conexión entre terminales mientras que la VoIP requiere apoyarse en un protocolo de control de sesión (señalización) con el que poder determinar factores clave como por ejemplo, la dirección física de la entidad a la que se desea llamar, datos relativos a la codificación de la sesión multimedia, etc. Todos estos aspectos serán tratados en mayor profundidad más adelante.

II. MOTIVACIÓN Y OBJETIVOS

1. Introducción

A continuación se tratará primero de introducir a la empresa promotora de este proyecto. Para ello se enmarcará su rango de actividad, se explicará qué servicios ofrece así como el funcionamiento y estructura geográfica del negocio.

1.1. Actualize

Se trata de una empresa multinacional dedicada a un subsector de las telecomunicaciones, concretamente a la asistencia informática remota a través de herramientas propias con un importante grado de personalización de acuerdo a las necesidades del cliente.

Servicios

Los clientes potenciales de Actualize son grandes empresas multinacionales interesadas en ofrecer a sus clientes servicios adicionales a los ya contratados. Estos servicios se basan fundamentalmente en extensiones del contrato con el cliente introduciendo productos de asistencia informática y tecnológica principalmente. Para ello Actualize crea y desarrolla aplicaciones exclusivas adecuadas a cada necesidad.

Estas aplicaciones son instaladas por los clientes finales en sus ordenadores de manera que ante cualquier incidencia relacionada con el ordenador esta pueda ser resuelta si no remotamente, in situ.

¿Cómo funciona?

El cliente, ante cualquier problema informático tiene dos maneras de contactar con Actualize para resolverlo. Puede hacerlo por medio de un chat o a través de un teléfono de contacto.

El Gráfico 1 muestra un esquema de los diferentes modos de los que dispone el cliente para establecer contacto con Actualize, siendo A la interfaz informática a través de la cuál se gestionan y escalan todas las llamadas.

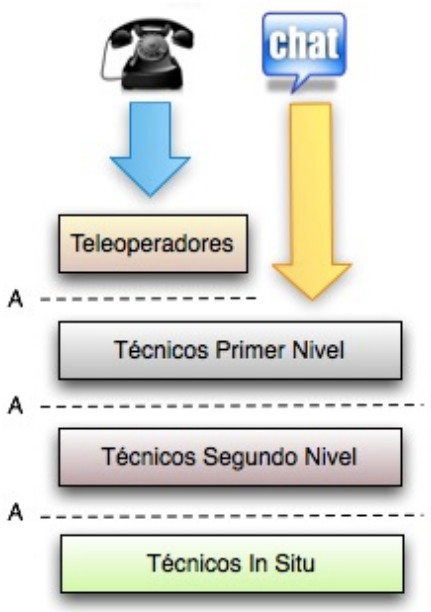


Gráfico 1: Esquema de recepción de llamadas

Dada la estructura citada anteriormente, podemos realizar una primera distinción de las diferentes partes implicadas en emisión y recepción de llamadas:

- **Teleoperadores** (TO): su trabajo consiste en recepcionar las llamadas que los clientes realizan e intentar, en caso de ser posible, solucionar el problema. Si el problema requiere un nivel mayor de cualificación la llamada es escalada a un Técnico de Primer Nivel. Adicionalmente, dan soporte a nuevos clientes que desean activar el servicio.

- **Técnico de Primer Nivel** (TPN): recibe avisos a través de la interfaz A (mirar Gráfico 1 con los clientes a los que debe llamar. A diferencia de los TO, los TPN emiten llamadas pero no reciben.

- **Técnico de Segundo Nivel** (TSN): disponen de la mayor cualificación técnica. Desarrollan las mismas funciones que los TPN con la diferencia de que el grado de dificultad de las incidencias es mayor. Al igual que sus antecesores, también emiten llamadas pero no reciben.

Cabe indicar que todos los anteriores deben poder emitir y recibir llamadas internas.

A continuación, para una mejor comprensión del flujo de llamadas y qué partes intervienen en su distribución, se expone un diagrama del proceso.

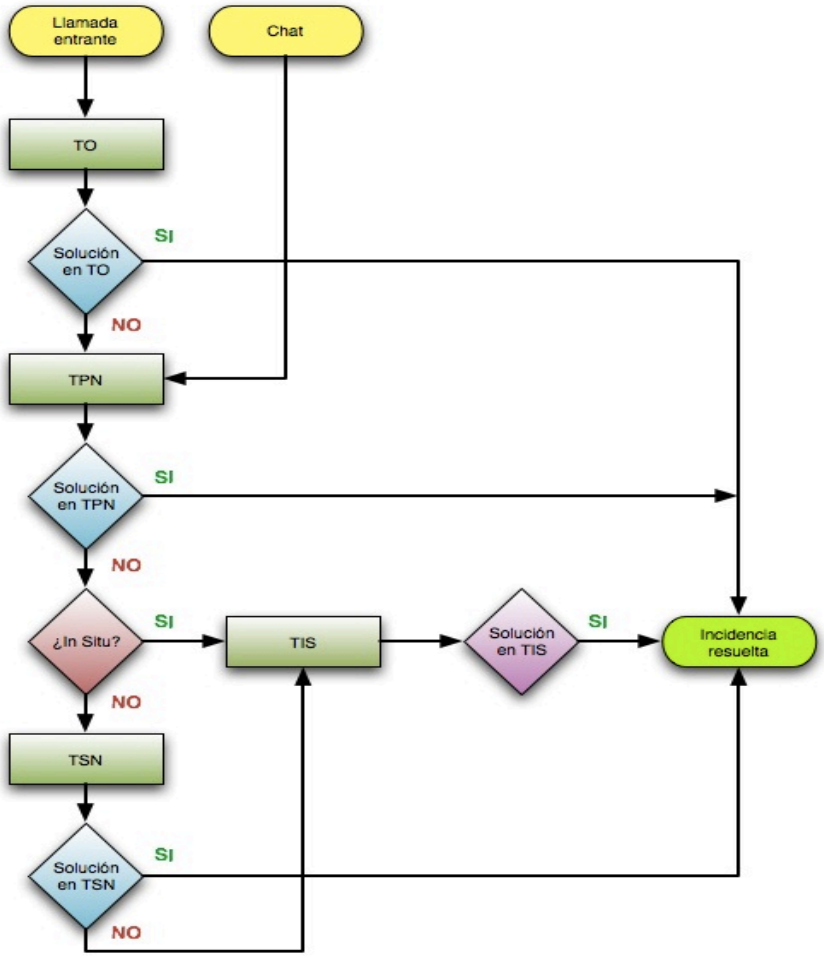


Diagrama 1: proceso de entrada de clientes finales

Estructura geográfica

Con el objeto de entender los motivos por los que se plantea este proyecto es necesario introducir una explicación de los diferentes elementos geográficos que forman la empresa y su conexión.

En un sistema socio-económico globalizado en el que las pretensiones de cualquier negocio son, entre otras, alcanzar la mayor cuota de mercado en su sector, es natural que la tendencia de una empresa como Actualize, dedicada a un subsector de las telecomunicaciones y servicios de valora añadido, requiera la expansión geográfica para lograr un mayor alcance.

Con sede en Madrid, la Empresa cuenta con cinco delegaciones, mayormente distribuidas en Latinoamérica (Colombia, Méjico, Brasil, Chile) e India. Cada delegación es responsable principalmente de manejar el tráfico de incidencias propio de cada país.

Gráfico 2: Distribución geográfica de Actualize

1.1.1. Motivaciones y objetivos

El proyecto objeto de este documento surge de la necesidad por parte de la empresa Actualize S.L. de migrar su sistema de comunicaciones telefónicas a un entorno *VoIP*. Los motivos motores del cambio son:

Operabilidad. Durante muchos años han existido las Centrales Privadas de Conmutación o PBX (*Private Branch Exchange*). Estos dispositivos son centrales telefónicas independientes que, además de gestionar las llamadas dentro de la empresa, dirigen las llamadas entrantes y salientes. Tradicionalmente, estas centralitas han sido poco transparentes y flexibles para el administrador de comunicaciones de la empresa. Poco transparentes dado su hardware y programación propietaria; poco flexibles ya que la información que almacenan

este tipo de centralitas es mínima dando lugar a una pobre capacidad de análisis y gestión del comportamiento telefónico[2] de la empresa.

Gastos. La manera en la que la empresa llama es vital para determinar una estrategia óptima con la que abaratar gastos.

Con el fin de dar un primer paso hacia el cambio de sistema de comunicaciones telefónicas, la empresa decidió contar conmigo para realizar un estudio preliminar de concepto y evaluar posibilidades. Dentro de los objetivos de este proyecto se contempla:

1) Adquirir habilidades teórico-prácticas en el uso del software libre (bajo licencia GPL: *General Public License*) Asterisk.
2) Familiarizarse con las tecnologías implicadas en el desarrollo y despliegue de entornos de comunicaciones con telefonía IP.
3) Instalar y configurar una centralita de telefonía VoIP (PBX) con funcionalidad básica de acuerdo a las necesidades de la empresa.
4) Capacidad de evaluar diferentes escenarios y arquitecturas junto con su viabilidad técnica para un posterior estudio de costes a fin de determinar la mejor solución VoIP para la empresa de acuerdo a sus previsiones.

[2] Se entiende comportamiento telefónico como los hábitos de uso de la empresa. **Cómo** llaman las personas que usan el sistema telefónico.

III. PLANTEAMIENTO DEL PROBLEMA

1. Introducción

Anteriormente se describieron algunos de los motivos por los cuales Actualize ha introducido dentro de su estrategia el cambio a VoIP. Una de las apuestas de la empresa es que parte de los trabajadores implicados en la resolución de incidencias (TO, TPN, TSN) desarrollen su actividad laboral desde su domicilio. En la actualidad una gran mayoría de este personal cumple ya con este punto y aun siendo el escenario ideal, debido al actual sistema de comunicaciones telefónicas, todavía no aporta todos los beneficios que debiera.

En años pasados, todos los técnicos y operadores desempeñaban su actividad desde las oficinas de la empresa. Cuando una llamada recepcionada por un teleoperador quería ser transferida a un técnico no había mayor problemática dado que ambos se encontraban dentro de las mismas instalaciones. Ahora, teniendo el mismo sistema de comunicaciones telefónicas basados en una PBX tradicional, no es posible transferir las llamadas a agentes que se encuentren en sus domicilios. Esto repercute de manera directa en la calidad del servicio ofrecido ya que el tiempo de respuesta es mayor (durante el proceso, al tener que colgar la llamada, el tiempo que transcurre desde que el cliente llama hasta que se le devuelve la llamada aumenta considerablemente. Esto da lugar a situaciones en las que el cliente ya no se encuentra disponible, se acumulan las llamadas pendientes y por consiguiente disminuye la prestación de servicio).

A lo largo de los siguientes apartados se introducirán las tecnologías implicadas en el proyecto con el fin de aportar una visión general para un posterior seguimiento en la lectura del diseño de la solución técnica. Seguidamente se expondrán los requisitos que requiere la solución.

2. Tecnologías implicadas y estado del arte

2.1. Asterisk

2.1.1. Introducción

Cuando hablamos de VoIP, la tendencia general es pensar en su uso para llamadas gratuitas. La realidad y el verdadero valor de la VoIP es el trato de esta como otra aplicación más en la red de datos.

Es necesario no olvidar el propósito del teléfono el cual se basa en hacer posible la comunicación entre personas. Hoy en día, es posible hacerlo de manera mucho más flexible y creativa gracias a tecnologías como Asterisk.

Cuando Asterisk se creó (1999) ya existían soluciones de telefonía VoIP basadas en software libre. Por aquel entonces también surgió el Proyecto de Telefonía Zapata (Zapata Telephony Proyect). Ambos, en conjunción, consiguieron crear el primer sistema con interfaz para PSTN trayendo consigo una revolución al integrar los dos mundos de la telefonía: VoIP y PSTN.

Al hablar de Asterisk hablamos de un software libre capaz de transformar un ordenador en un servidor de comunicaciones. Al tratarse de un proyecto de software libre cuenta con una enorme comunidad de desarrolladores y usuarios. Las grandes marcas de centralitas telefónicas propietarias crean productos en los que participan un pequeño grupo de desarrolladores. Es complicado que este grupo reducido de personas tenga el mismo volumen de ideas que pueden tener millones de usuarios alrededor del mundo. Dado que no es posible modificar el software de estos productos su evolución es capada. Esto no ocurre con el software libre y más concretamente con Asterisk. La gente puede proponer, desarrollar e implantar nuevas ideas que de otra manera tardarían mucho tiempo más en llegar a nuestras manos. Es aquí donde reside el potencial de Asterisk.

Quizá uno pueda pensar que es irresponsable por parte de una empresa basarse en un software que no está controlado por “nadie” pero controlado por “todo el mundo” al mismo tiempo. Esta idea no es acertada y hay ejemplos que lo corroboran. El más famoso probablemente es Linux. Millones de empresas de todo el mundo usan en este sistema operativo no propietario obteniendo grandes resultados. Esta gran aceptación por parte de la industria ha facilitado el camino para la evolución e implantación de Asterisk.

Cuando un nuevo usuario se incorpora a la utilización de un nuevo sistema como es Asterisk requiere soporte e información. Asterisk cuenta con múltiples formas de aportar este requerimiento. En primer lugar cuenta con una gran comunidad realmente colaborativa dispuesta a solucionar cualquier problema. Su *mail list* distribuye varios cientos de mensajes al día y por si esto fuera poco, existen sitios tipo *wiki* con gran contenido como voip-info.org, canales de IRC, Grupos de Usuarios de Asterisk, etc.

2.1.2. Arquitectura

Existe una gran diferencia entre el plan de marcado de Asterisk y el de las PBX tradicionales. En las PBX tradicionales existe una diferencia lógica entre las estaciones (teléfonos) y los trunks (recursos para tener conectividad con el mundo exterior) mientras que Asterisk trata todos los canales de la misma manera.

En Asterisk, todo lo que viene del exterior pasa por algún tipo de canal. Aunque hay varios tipos de canales, el *dialplan*[3] trata todos los canales de manera similar de manera que por ejemplo, un usuario podría salir por un trunk hacia el exterior y ser tratado por el dialplan de la misma manera que lo hace con un usuario interno.

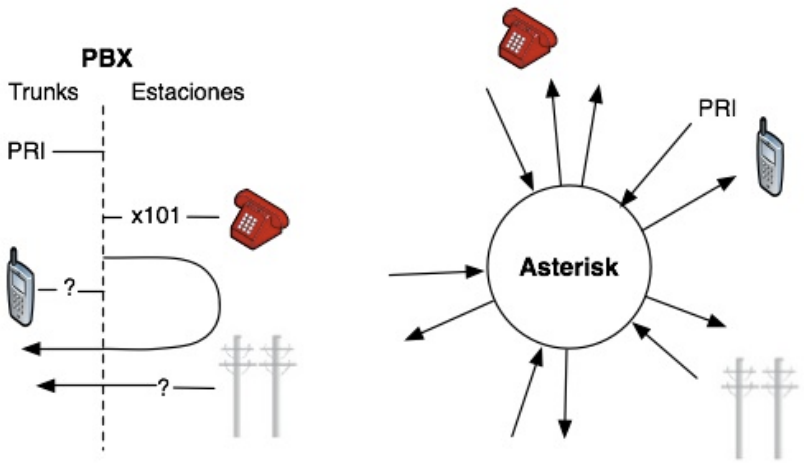


Gráfico 3: Arquitectura Asterisk vs PBX tradicional

2.1.2.1. Módulos

Asterisk está formado por módulos. Un módulo es un componente cargable que aporta una funcionalidad específica. Los módulos de Asterisk son cargados en base al archivo */etc/asterisk/modules.conf.* Comentar que es posible iniciar Asterisk sin ningún módulo cargado y cargarlos manualmente por consola a medida que queramos añadir funcionalidad. Esto no es recomendable y tiene sentido en el caso de que queramos aumentar el rendimiento del sistema y desactivar aquellos módulos que no cumplen ninguna función.

[3] Traducido al español, *dialplan*, se podría expresar como "plan de marcado". En esta memoria se le llamará *dialplan* debido al extenso uso de esta palabra.

Existen diferentes tipos de módulos:

- **Aplicaciones**: Usadas en el archivo extensions.conf para definir las distintas acciones que pueden ser aplicadas a una llamada.
- **Módulos puente**: Cumplen funciones de multiplexado para conferencias, puente entre canales, etc.
- **Módulos de registro de detalle de llamada (CDR)**: proporcionan múltiples métodos con los que registrar los detalles de llamada.
- **Módulos de registro de eventos de canal (CEL)**: Provee mayor control sobre el reporte de actividad de llamada.
- **Drivers de canal**: Sin los drivers no sería posible realizar llamadas. Estos son la puerta de enlace hacia el *core* de Asterisk.
- **Traductores de códec**: permite hacer las conversiones que sean necesarias para traspasar llamadas de un canal a otro.
- **Interpretadores de formato**: misma función que los traductores de códec con la diferencia de que los interpretadores trabajan sobre archivosen vez de sobre canales.
- **Funciones del dialplan**: complementan a las aplicaciones proporcionando mejoras como el manejo de strings por ejemplo.
- **Módulos de PBX**: son módulos periféricos que aportan mejor control y mecanismos de configuración.
- **Módulos de recursos**: integra Asterisk con recursos externos.
- **Módulos adicionales o addons**: módulos desarrollados por la comunidad que rodea Asterisk. No son cargados por defecto.
- **Módulos de test**: Se trata de módulos empleados por la gente que desarrolla software para Asterisk para validar nuevo código.

2.1.2.2. Estructura de archivos

Archivos de configuración

Algunos de los archivos de configuración más importantes son el *extensions.conf, sip.conf, modules.conf* entre otros. Todos ellos se encuentran en el directorio */etc/asterisk.*

Módulos

Normalmente son instalados en el directorio */usr/lib/asterisk/modules.* Normalmente no es necesario trabajar con esta carpeta pero puede ser de utilidad para actualizaciones por ejemplo.

Librería de recursos

Algunos recursos necesitan usar fuentes de datos externos. Estas pueden ser archivos de audio para buzón de voz por ejemplo o scripts de diversa índole. El directorio se donde estos se almacenan es *var/lib/asterisk.*

Spool

En Linux el *spool* es donde se almacenan archivos que varían frecuentemente. Es aquí (*/var/spool/asterisk)* donde Asterisk guarda los mensajes de voz, grabaciones de llamada, etc.

Logging

Cualquier tipo de registro como puede ser los generados por el módulo CDR o el CEL, mensajes, errores son guardados en */var/log/asterisk.* Este directorio cobra gran importancia a la hora de enfrentarse a la resolución de problemas del sistema.

Dialplan

Es probablemente la parte más importante de Asterisk ya que es quien proporciona toda la lógica de marcado y manejo de llamadas.

Aunque existen tres tipos de sintaxis para escribir el dialplan, la más popular sin duda alguna es la sintaxis "tradicional" de Asterisk. El archivo donde emplear esta sintaxis es */etc/asterisk/extensions.conf.*

2.1.2.3. Conceptos importantes

Para comprender el funcionamiento y potencial que ofrece Asterisk es necesario explicar algunos conceptos previos.

Extensiones

En Asterisk, el término extensión no tiene el significado al que estamos acostumbrados. En Asterisk, una extensión es un patrón marcado por algún dispositivo. Cuando Asterisk recibe este patrón, es procesado por el dialplan realizando las acciones que se hayan especificado para ese patrón marcado. Por ejemplo, un softphone puede marcar la extensión 100. Cuando se marca esta extensión, una PBX tradicional contactará el teléfono asignado a dicha extensión. Sin embargo, si es marcada en un dispositivo conectado a Asterisk, este puede realizar la misma función que una PBX tradicional y contactar con un usuario pero podría también llamar al buzón de voz, reproducir un mensaje o

incorporarse a un cuarto de conferencia. Esta forma de tratar las extensiones es mucho más flexible que la forma tradicional.

Nombre de dispositivo o usuario

El nombre de dispositivo o usuario es el nombre con el que se registrará un dispositivo en Asterisk. Aunque existen recomendaciones pude usarse cualquier nombre siempre y cuando sean caracteres alfanuméricos y sin espacios.

- ✓ *Softphone*: se trata de programas que son utilizados en un ordenador o en un dispositivo móvil como son los smartphones.
- ✓ *Hardphone* (teléfono VoIP): dispositivo físico con el mismo aspecto que un teléfono tradicional. Se conecta directamente a la red
- ✓ *Analog* Terminal Adaptors (ATAs): puertas de enlace analógico-digital para teléfonos analógicos tradicionales.

2.1.3. Interfaces web

La interfaz web por excelencia es FreePBX. Se trata de software libre y se encuentra en el corazón de muchas distribuciones como AsteriskNow, PBX in a flash and Trixbox. Todas ellas utilizan en mayor o menor medida FreePBX, una interfaz gráfica con la que cualquier persona, sin grandes conocimientos técnicos (Linux), puede aprovecharse de toda la funcionalidad que aporta el motor software Asterisk.

Sin embargo, aunque aparentemente pueda parecer que uno no necesita saber cómo funciona tanto Asterisk como su sintaxis y su estructura, lo cierto es que tener experiencia con ello ayuda en la comprensión y resolución de problemas que pueden surgir a la hora de trabajar con FreePBX.

2.2. Protocolos en VoIP

Que el sistema VoIP se haya desarrollado con base en el protocolo IP tiene sentido. La comunicación telefónica podría ser identificada como una necesidad en la sociedad en la que vivimos. Esta necesidad nace, en parte, de la enorme difusión de la red telefónica y el fácil acceso a una línea telefónica. El protocolo IP parece cumplir con esta premisa ya que es el protocolo usado para que millones de personas en todo el mundo se conecten a Internet.

Al igual que en una Red Telefónica Pública Conmutada los conmutadores dirigen una llamada, el protocolo IP tiene la función de establecer las normas para que los enrutadores consigan hacer llegar un paquete origen hasta su destino.

La pregunta que cabría hacerse es si existe la necesidad del uso de protocolos específicos para este tipo de redes (VoIP).

La transmisión de voz o la mayoría de flujos de datos multimedia requieren unas mínimas prestaciones para que la comunicación sea satisfactoria. Estas prestaciones se traducen principalmente en la ordenada llegada de los datos al destino y dependiendo del tipo de flujo multimedia y la aplicación[4], un bajo retardo es deseable. En una red con un sistema de conmutación de circuitos (RTPC) el retardo no es un problema ya que este se encuentra acotado mientras que en conmutación de paquetes (Internet) esto no ocurre siendo este variable.

Dada la pretensión del sistema VoIP (igualar en cuanto a calidad de servicio al sistema telefónico tradicional con todas sus implicaciones) y la naturaleza de las redes IP (conmutación de paquetes) es necesario incluir nuevos protocolos que ayuden a solucionar o al menos mitigar los problemas derivados de trabajar en una red como Internet.

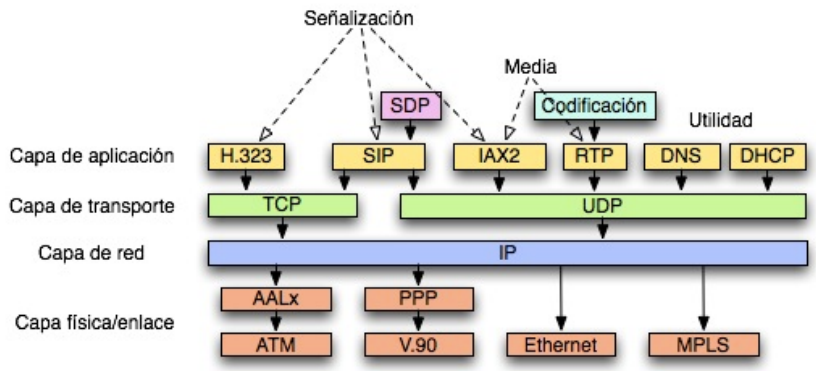


Gráfico 4: Pila de protocolos

[4] Es necesario hacer una distinción entre los diferentes tipos de datos o flujos multimedia y sus requerimientos a la hora de ser transportados ya que sus aplicaciones son diferentes. Por ejemplo, los datos de voz pueden ser los mismos en una grabación (archivo de audio) y en una conversación en tiempo real pero no tienen las mismas necesidades. Al transportar datos de voz pertenecientes a un archivo de audio la latencia cobra poca importancia ya que los datos terminarán llegando a su destino. Sin embargo, para que una conversación entre dos personas pueda tener lugar, el retardo de los paquetes es un punto clave para que esta pueda tener lugar.

2.2.1. Session Initiation Protocol (SIP)

Las siglas SIP hacen referencia al Protocolo de Inicio de Sesión del inglés *Session Initiation Protocol*. Fue desarrollado por *el Multy-Party Multimedia Session Control Working Group* del IETF. Su misión es establecer las pautas y mecanismos para la iniciación, modificación y finalización de sesiones multimedia como son la voz, el vídeo, la mensajería instantánea o información de presencia.

Existen gran variedad de parámetros y variables que deben ser acordados entre las partes de una comunicación antes de establecerse una sesión multimedia. Para ello, el protocolo SIP hace uso de protocolos "auxiliares" y herramientas de manera que cada protocolo tiene funciones específicas.

El gráfico 3 muestra la pila de protocolos sobre los que se monta el propio SIP así como otros protocolos usados en la señalización de comunicaciones multimedia (H.323, IAX2).

Tal como se ha comentado, SIP es un protocolo dedicado a la señalización y no al transporte del flujo multimedia. El encargado de esto último es el protocolo **RTP** (*Real-Time Transport Protocol*).

Real-Time Transport Protocol

Este protocolo sirve como medio de transporte de red *end-to-end*[5] para aplicaciones que requieren la transmisión de información en tiempo real tales como audio o video. Su función no cubre la reserva de recursos y no garantiza calidad de servicio (QoS) para servicios en tiempo real.

Con el fin de aumentar la funcionalidad y las prestaciones, RTP hace uso del protocolo **RTCP** (*Real-Time Transport Control Protocol*), encargado de aportar información relativa a la calidad de servicio prestada por RTP. Entre los parámetros que maneja se encuentran los bytes y paquetes enviados, paquetes perdidos o *jitter* entre otros. De acuerdo a los resultados que refleje RTCP pueden tomarse contramedidas destinadas a aumentar la calidad de servicio como por ejemplo cambio de códecs de compresión.

[5] Principio que estipula que las funciones específicas de aplicación de un sistema debieran estar en los hosts finales y no en los nodos intermedios.

Session Description Protocol

A la hora del inicio de una sesión multimedia es necesario acordar detalles relativos al flujo media entre las partes así como las direcciones de transporte y otros metadatos de descripción de sesión útil para los participantes.

SDP brinda un formato de representación estándar para la comunicación de este tipo de detalles independientemente de la forma de transporte (puede ser usado conjuntamente con diferentes protocolos de transporte como SAP (Session Announcement Protocol), SIP (Session Initiation Protocol), RTSP (*Real Time Streaming Protocol*), correo electrónico usando extensiones MIME, y *Hypertext Transport Protocol*.

Elementos de una red SIP

SIP es un protocolo de naturaleza cliente-servidor. A continuación se presentan las distintas entidades que existen en una red SIP.

Agentes de Usuario (UA)

Se denomina agente de usuario (UA) a cualquier dispositivo final que esté habilitado para trabajar con SIP. Como el propio nombre indica, un agente de usuario es el encargado de recoger información del usuario y ejercer como agente para establecer o terminar sesiones SIP con otros agentes. Aunque lo normal es que el usuario sea una persona, también puede ser un protocolo.

Debido a que SIP puede ser usado mediante cualquier protocolo de transporte, debe soportar TCP y UDP. Según el comportamiento del agente de usuario, este puede ser cliente (UAC: User Agent Client) o servidor (UAS: User Agent Server). Un agente de usuario es UAC cuando realiza una petición y UAS cuando la recibe.

Agentes de presencia (PA)

Un agente de presencia es un dispositivo SIP capaz de recibir peticiones de suscripción y generar notificaciones de estado tal como se definen en la especificación de Eventos SIP.

Un agente de presencia puede recolectar información de presencia de varios dispositivos. Esta información de presencia puede venir de un registro por parte de un dispositivo SIP, un dispositivo SIP publicando información de presencia o incluso de otras fuentes que no sean SIP.

Existen los denominados servidores de presencia los cuales, dependiendo de las condiciones, actúan como agentes de presencia o como un proxy reenviando peticiones de SUBSCRIBE[6] a otro agente de presencia.

Agentes de usuario Back-To-Back (B2BUA)

Un agente de usuario Back-to-Back es un dispositivo SIP el cual recibe una petición SIP, la reformula, y la reenvía como una nueva petición. Las respuestas a estas peticiones son también reformuladas y reenviadas en la dirección opuesta.

Este tipo de entidades son un punto de fallo en una red SIP lo que significa que su uso puede reducir la fiabilidad de una sesión SIP en Internet.

La forma más común de B2BUA son puertas de enlace de nivel de aplicación (Application Layer Gateways, ALG) los cuales se encuentran en algunos *firewalls* con el (a veces no tan satisfactorio[7]) propósito de habilitar las sesiones SIP sin reducir por ello la seguridad.

[6] Este tipo de mensajes será explicado en posteriores apartados.

[7] Uno de los mayores problemas a la hora de establecer comunicaciones SIP es la intrusión de los *firewalls* en el flujo SIP a través de diferentes mecanismos como pueden ser las ALGs. Aunque la pretensión por parte de los fabricantes es ayudar en el establecimiento de comunicaciones SIP, lo cierto es que muchos de estas soluciones "rompen" los paquetes SIP impidiendo una correcta comunicación.

Puertas de enlace

Una puerta de enlace SIP es una aplicación encargada de ser la interfaz entre una red SIP y cualquier otra red que utilice protocolos de señalización diferentes. En términos de protocolo, una puerta de enlace SIP es otro tipo de agente de usuario que en vez de actuar en lugar de una persona como usuario actúa "representando" otro protocolo.

Servidores

Un servidor SIP es una aplicación que acepta peticiones SIP y responde a ellas. Es importante no confundir este tipo de servidores con los servidores agente de usuario (UAS) o la naturaleza cliente-servidor del protocolo. Un servidor SIP es una entidad lógica y un servidor físico puede contener varios servidores SIP o incluso operar con uno u otro dependiendo de que se den unas condiciones u otras. Dado que los servidores prestan servicio a los agentes de usuario, deben soportar TCP y UDP para transporte.

Servidor proxy

Un servidor proxy no es un B2BUA. Un servidor proxy SIP recibe peticiones de un agente de usuario u otro proxy y actúa de parte del agente de usuario en reenviando o respondiendo a la petición. Es importante resaltar que no es un B2BUA porque un servidor proxy sólo puede modificar las peticiones y respuestas siguiendo unas pautas concretas descritas en el RFC 3261.

Para poder reenviar las peticiones, el servidor proxy necesita una base de datos de algún tipo donde pueda consultar las direcciones pertenecientes a los agentes de usuario. Resaltar que la interfaz entre este servicio de localización y el proxy no está definida en el protocolo SIP.

Un servidor proxy se diferencia de un agente de usuario o puerta de enlace en tres puntos clave:

- Un servidor proxy no emite peticiones. Sólo responde a peticiones de un agente de usuario (la petición `CANCEL` es una excepción a esta regla)
- Un servidor proxy no tiene capacidades para media.

- Un servidor proxy no analiza el cuerpo de los mensajes; solamente se basa en cabeceras.

A continuación se muestra un tipo de topología de red común llamada Trapecio SIP dada la figura que forman los mensajes de señalización y los de media.

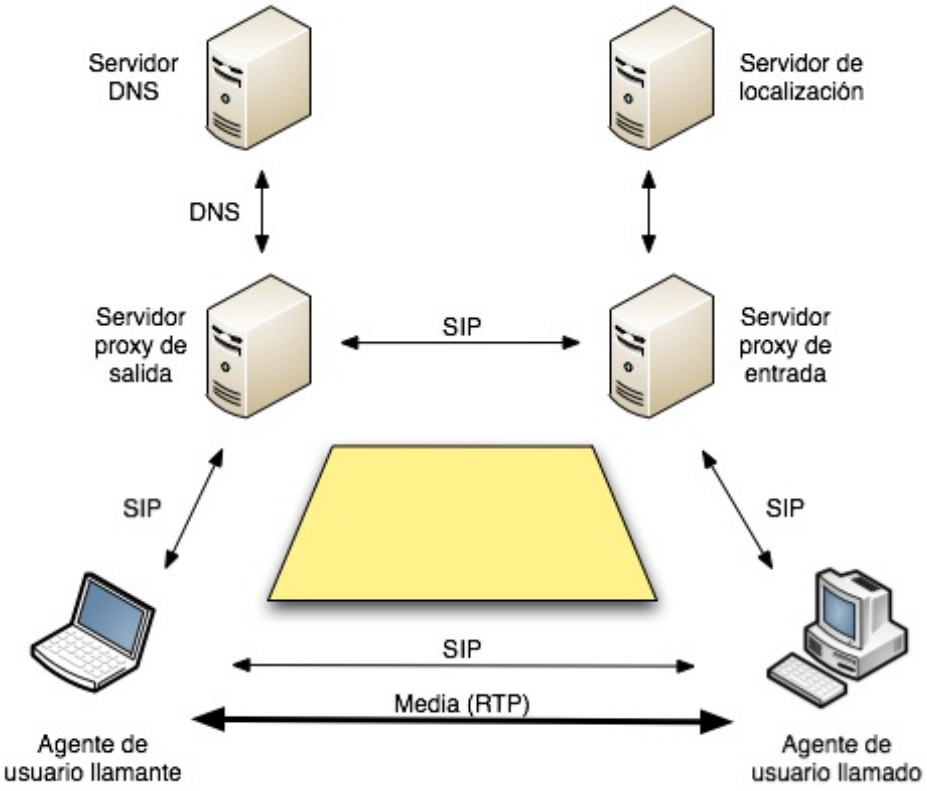


Gráfico 5: Trapecio SIP

Al mismo tiempo, un servidor proxy SIP puede ser de dos tipos, sin estado o con estado:

Servidor proxy sin estado (stateless)

Un servidor sin estado o stateless procesa cada petición o respuesta SIP basándose solamente en el contenido de los mensajes. Esto quiere decir que una vez el mensaje ha sido analizado, procesado y enviado no se almacena ningún tipo de información relativa al mensaje. Por otra parte, un servidor de estas características nunca retransmite un mensaje y no usa contadores de tiempo SIP.

Para poder identificar *loops* en la transmisión de mensajes se implementa un campo obligatorio en todas las peticiones llamado `Max-Forwards.`

Servidor proxy con estado (stateful)

Este tipo de servidores almacenan información tanto de las peticiones como de las respuestas recibidas para usarla en el procesado de futuras peticiones y respuestas. Por ejemplo, un servidor proxy con estado inicia un contador de tiempo cuando una petición es reenviada. Si no se recibe una respuesta dentro del periodo de tiempo estipulado el servidor volverá a reenviar la petición, quitando al agente de usuario la responsabilidad de hacerlo.

Hay un tipo de servidor proxy con estado llamado **proxy con estado de transacción (*transaction stateful proxy*)**. Este, el más común, mantiene información de una transacción pero sólo mientras la petición está pendiente. Es decir, sólo hasta que se recibe una respuesta a esa petición. Después de recibir la respuesta, sea del tipo que sea, la información de estado es destruida.

Servidor de redirección (redirect)

Un servidor de redirección responde peticiones pero no las reenvía. Tal como hace el servidor proxy, este tipo de servidores también usa una base de datos o un servicio de localización para poder averiguar la dirección de los usuarios. Esta información es enviada de vuelta al llamante mediante una mensaje de respuesta de clase redirección (*redirection class response*) (3xx).

Servidor de registro (registrar)

Este tipo de servidores, también llamados registradores o *registrar,* acepta peticiones SIP de tipo `REGISTER`. Cualquier otra petición que se envíe al servidor recibirá como respuesta un mensaje de tipo `501 Not Implemented.` La información de contacto de la petición es entonces accesible para cualquier otro servidor SIP dentro del dominio de administración tales como *proxies* y *registers.*

Por seguridad, el servidor de registro requiere que el agente de usuario que está siendo registrado se autentique para que las llamadas realizadas a este no sean interceptadas o desviadas (alguien no autorizado podría hacer que el URI SIP señalase a su propio teléfono). [8]

El mensaje tipo `REGISTER` brinda varias posibilidades al agente de usuario según los campos de cabecera que contenga. Entre ellas, puede hacerse con una lista de los registros actuales, limpiar el registro, o añadir un nuevo registro URI a la lista.

En el siguiente gráfico se expone de manera sencilla la interacción entre varios elementos anteriormente citados.

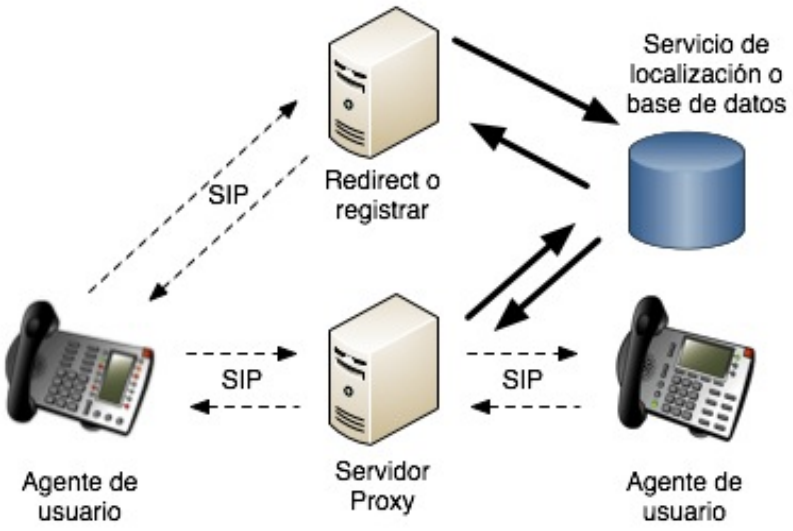


Gráfico 6: Interacción entre UA, servidores y servicio de localización

Los mensajes SIP pueden ser de dos tipos:

- peticiones o *requests*, y
- respuestas o *responses*

La composición de los mensajes consiste en una línea inicial (indica si se trata de una petición o una respuesta) seguida de uno o más campos de cabecera (headers), una línea vacía indicadora de final de las cabeceras y por último y de manera opcional, el cuerpo del mensaje.

[8] Para mantener los niveles de seguridad óptimos es conveniente usar TLS (Transport Layer Security) ya que HTTPS Digest no proporciona toda la seguridad requerida.

Todos los mensajes de una transacción SIP cuentan con unas cabeceras obligatorias:

- **Via**: Indica el modo de transporte (UDP o TCP) así como la ruta del mensaje (cada entidad SIP por la que transcurre el mensaje deja su dirección IP)
- **From**: Dirección de origen del mensaje
- **To**: Dirección del destinatario al que se destina el mensaje
- **Call-Id**: Identificador de llamada (único para cada llamada). Este debe ser el mismo para todos los mensajes de una misma transacción.
- **Cseq**: Número decimal que es incrementado en cada petición (normalmente) en 1.

Mensajes de petición SIP

Aunque son seis los métodos principales incluidos en el RFC 3261, existen otros métodos de menor protagonismo definidos en diversos documentos RFC diferentes al principal.

Métodos

Una petición SIP o método puede ser considerado como el "verbo" dentro del protocolo ya que su misión es pedir que una cierta acción sea tomada por otro agente de usuario o un servidor. Tal como indicábamos en el párrafo anterior, seis son los métodos principales: `INVITE, REGISTER, BYE, ACK, CANCEL, y OPTIONS`. Por otro lado, en RFCs diferentes, encontramos los métodos `REFER, SUBSCRIBE, NOTIFY, MESSAGE, UPDATE, INFO, y PRACK.`

- **INVITE**: Es el método usado para establecer sesiones multimedia entre agentes de usuario. El acuse de recibo (*acknowledge*) de las respuestas al INVITE se realiza mediante el método ACK que será descrito más adelante. Normalmente este método contiene información relativa a la sesión multimedia aunque pueden contener otro tipo de información como aspectos relacionados con la calidad de serivicio (QoS) .
- **REGISTER**: Es usado por un agente de usuario para notificar a una red SIP su actual `Contact` URI (dirección IP) y la URI cuyas peticiones deben ser enrutadas a este `Contact`.

- **BYE:** Este mensaje es utilizado para terminar una sesión SIP. Este tipo de mensajes sólo son enviados por agentes de usuario que participen en la sesión y nunca por proxies u otros dispositivos.
- **ACK:** El método ACK se utiliza para acusar recibo de respuestas finales correspondientes a un mensaje INVITE. Notar que respuestas a cualquier otro método no son acusadas. El método ACK puede contener información de descripción de sesión (SDP) en el cuerpo del mensaje. Esto es permitido si el INVITE inicial no contiene esta información.
- **CANCEL:** Como el propio nombre del método indica sirve para cancelar o terminar búsquedas pendientes o intentos de llamada.
- **OPTIONS:** Utilizado para preguntar a un agente de usuario o servidor sobre sus capacidades y averiguar su disponibilidad actual. Un proxy nunca puede generar este tipo de mensajes.

Una característica importante ya citada es la transparencia de los servidores proxy a la hora de tratar con peticiones. Este tipo de servidores no necesita comprender un método de petición para poder reenviarlo. De hecho, cualquier método no conocido es tratado como un OPTIONS. De esta manera, es posible introducir nuevas características y métodos útiles para los agentes de usuario y sin embargo no es necesario "actualizar" los proxies y que estos los comprendan.

Mensajes de respuesta SIP

Las respuestas SIP son mensajes generados por un agente de usuario servidor (UAS) o un servidor como respuesta a una petición generada por un agente de usuario cliente (UAC).

Existen seis clases de respuestas SIP. Las primeras cinco fueron "prestadas" de HTTP; la sexta fue creada exclusivamente para SIP. En la Tabla se muestran los diferentes tipos de respuestas.

Clase	Descripción	Acción
1xx	Informativa	Indica el estado de la llamada antes de ser terminada
2xx	Éxito	Petición aceptada
3xx	Redirección	Servidor ha devuelto posibles localizaciones. El cliente debe probar petición en otro servidor.
4xx	Error de cliente	La petición ha fallado debido a un error en el cliente. El cliente puede reintentar la petición si esta es reformulada acorde a la respuesta.

5xx	Fallo del servidor	La petición ha fallado debido a un error en el servidor. La petición puede ser reenviada a otro servidor.
6xx	Fallo global	La petición ha fallado. La petición no debe ser reenviada ni a este ni a otros servidores.

Tabla 1: Clases de respuestas SIP

2.2.1.1. Ejemplo flujo SIP

Una vez han sido revisados los conceptos básicos del protocolo SIP y sus diferentes implicaciones con otros protocolos nos encontramos en disposición de analizar el flujo de una transacción SIP identificando las diferentes partes implicadas así como la lógica de su funcionamiento.

A continuación se presentan varios ejemplos de transacciones SIP en los que intervienen diferentes entidades.

Establecimiento de sesión entre dos agentes de usuario

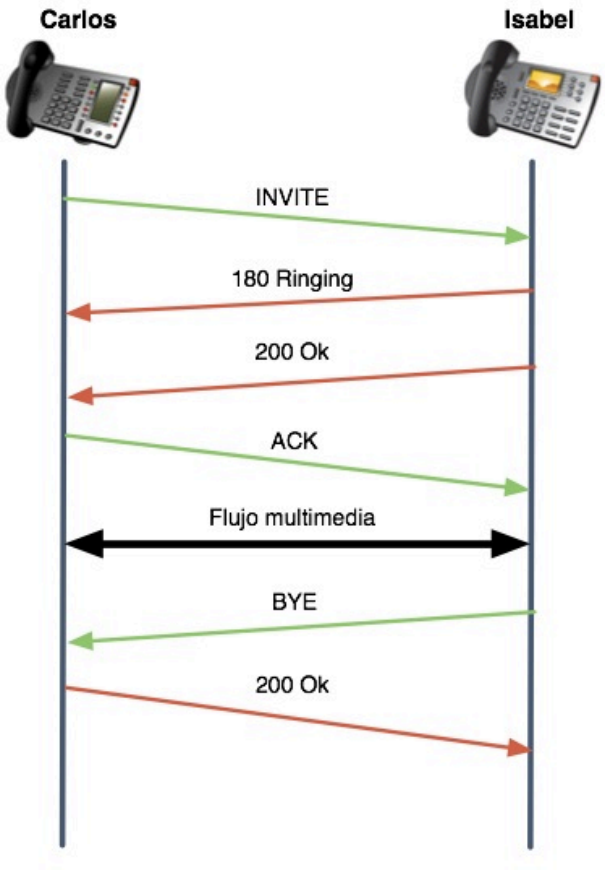


Diagrama 2: Sesión entre dos UA

En el ejemplo representado en el diagrama superior muestra el intercambio de mensajes SIP entre dos dispositivos que "hablan" SIP. Asumimos que ambos dispositivos están conectados a Internet y que conocen sus respectivas direcciones IP.

Carlos, el que llama, comienza mandando un mensaje de petición INVITE a la que es llamada, Isabel. El INVITE contiene los detalles de el tipo de sesión que se está solicitando (SDP). Esta puede ser de voz (audio), multimedia (videoconferencia por ejemplo), o podría ser una sesión de juego online.

El siguiente mensaje es una respuesta al INVITE por parte de Isabel (180 Ringing). Este mensaje indica que la petición ha sido recibida correctamente y que se está alertando (si se trata de una sesión de audio, el teléfono SIP sonaría).

Una vez la llamada en este caso es aceptada (se descuelga el teléfono), se envía otra respuesta del tipo 200 Ok (clase "Éxito" del mensaje).

Finalmente, Carlos manda un acuse de recibo (ACK) indicando a Isabel que la respuesta ha sido recibida correctamente. A partir de este instante el intercambio de media (audio en nuestro caso) tiene lugar utilizando para ello el protocolo definido en el mensaje SDP que contiene la petición INVITE y la respuesta 200 Ok.

Cualquiera puede terminar la sesión. En este caso es Isabel quien lo hace, mandando un mensaje de petición BYE, al cual contesta Carlos con un 200 Ok. La sesión es terminada.

2.2.1.2. SIP y NAT

Hoy en día podríamos no equivocarnos si asumimos que toda empresa dispone de un firewall entre su intranet e Internet. La función que este tipo de dispositivos aportan es seguridad frente a diferentes tipos de ataques o accesos no autorizados. Tanto es así que son muchos los *routers* o *hubs* inalámbricos que incorporan uno siendo también costumbre su uso en PCs personales.

Podemos pensar en un firewall como una puerta de un solo sentido en el que se deja pasar tráfico desde una intranet hacia Internet pero no a la inversa a no ser que se trate de respuestas a peticiones. Esto no es del todo cierto, ya que

si se admiten algunas peticiones siempre y cuando vayan dirigidas a servicios concretos como un servidor web mediante HTTP, un servidor de correo, etc.

Los buenos propósitos con los que los firewall fueron diseñados en pro de la seguridad también afectan a las sesiones SIP causando problemas a la hora de atravesarlos. Aunque en principio cualquier petición SIP que venga desde "fuera" es bloqueada este es le menor de los problemas ya que esto es solucionable con tan sólo abrir el conocido puerto que usa SIP (5060). El problema reside en que el tráfico media, que usa el protocolo RTP sobre UDP, utiliza varios puertos (aleatorios) siendo bloqueado por el firewall. Hay varias soluciones a este problema. Una de ellas consiste en que el firewall "entienda" SIP y sea capaz de analizar las peticiones INVITE y 200 OK provenientes de Internet, extraer los números de los puertos especificados en el SDP dentro de las peticiones y abrir "*pin holes*[9]" para que este tráfico pueda pasar. Otra alternativa es una ALG (Application-Level Gateway), que no es más que un B2BUA en el que el firewall confía. El firewall deja paso al tráfico SIP y RTP, terminando este en la ALG, y bloquea todo el tráfico restante. Es la ALG la que controla las sesiones SIP y no el firewall.

Otro gran problema a la hora de establecer sesiones SIP son los NAT (*Network Address Translator*). Un NAT, traducido como "Traductor de Direcciones de Red", es usado para conservar direcciones IPv4, y para ocultar las direcciones IP y la estructura LAN que está detrás del NAT. Tal como indica el nombre, un NAT traduce direcciones IP. Dada la escasez de direcciones IPv4 únicas existen direcciones no-únicas (llamadas privadas) del tipo 192.168.X.X, 172.16.X.X-172.29.X.X o 10.X.X.X . Estas son usadas dentro de las redes de área local. El encargado de traducir unas (privadas) en otras (pública) es el NAT.

Dado el funcionamiento de SIP al enrutar respuestas, un dispositivo SIP detrás de un NAT insertará en el campo de cabecera `Via` de cada uno de los mensajes originados su dirección privada. Cuando una petición es enviada fuera de una red local, las direcciones IP de las cabeceras IP y UDP serán cambiadas por la IP pública temporal. El NAT almacena la relación de direcciones IP públicas y privadas de las diferentes conexiones de tal manera que, cuando llegue una respuesta, la dirección IP se cambiará nuevamente de la pública a la privada y el paquete de enrutará correctamente.

[9] "pin en inglés significa alfiler, de ahí la expresión "pin holes": abrir pequeños agujeros en el firewall para facilitar la entrada de peticiones.

El problema reside en que los campos de cabecera `Via` , `Contact` y las direcciones IP del mensaje SDP no son modificadas. Para la señalización SIP esto no es un gran problema. Cada proxy o agente de usuario que recibe una petición compara la dirección IP del paquete con la dirección que figura en el campo `Via`. Si no coinciden es que existe un NAT entre medias. Para solucionar esto se añade un una etiqueta `received` en la que se indica la dirección IP pública. Fuera del NAT, la respuesta se enruta usando la dirección que figura en la etiqueta `received`. Una vez dentro de red local se utiliza la dirección IP del `Via`.

Sin embargo, esta solución no es válida para el flujo RTP. El puerto que se especifica en el mensaje SDP correspondiente a RTP es el puerto del cliente dentro de la red local no teniendo por qué ser el mismo en el NAT. Esto quiere decir que los paquetes RTP se destinarán a la dirección adecuada pero a un puerto diferente al que corresponde.

2.2.2. Otros protocolos

2.2.2.1. H. 323

El protocolo H.323 fue diseñado por la ITU (*International Telecommunication Union*) para poder transportar por IP datos pertenecientes a una videoconferencia. Aunque tuvo fama como protocolo para VoIP, se trata de un estándar para el equipamiento de videoconferencia basado en IP. Aunque probablemente siga siendo el protocolo para VoIP más usado por las portadoras, lo cierto es que su uso está en decremento en favor de protocolos como SIP o IAX[10] (*Inter Asterisk eXchange*).

2.2.2.2. IAX

Inter Asterisk eXchange. Este protocolo fue ideado por Digium, la empresa fundada por el (primer) desarrollador de Asterisk, como protocolo para que varios Asterisk se comunicaran entre sí. Aun siendo este el propósito de su creación, es importante destacar que no está limitado sólo a Asterisk; el estándar está abierto a aportaciones. IAX usa un solo puerto UDP (4569) tanto para la señalización como para el transporte del flujo *media.* Esto hace que su uso facilite en gran medida el paso por NAT.

[10] Aunque existe IAX e IAX2 ya no se presta soporte para la primera versión (IAX) y se utiliza el nombre genérico IAX para referirse a IAX2.

Otra gran ventaja de este protocolo es la posibilidad de "juntar" varias sesiones en un solo flujo de datos, lo que se traduce en un gran ahorro de ancho de banda (varias sesiones comparten cabecera).

Aunque IAX ha sido diseñado para VoIP, se trata de un protocolo abierto pudiendo ser implementado el transporte de cualquier otro tipo de flujo de datos (vídeo por ejemplo).

2.3. Codecs de compresión de voz

Aun habiendo quedando hoy en día algunos sistemas analógicos la realidad es que la gran mayoría son digitales. La "voz" que transcurre por las redes (todas las redes) lo hace a través de flujos de bits. La codificación de voz es el proceso por el cual se representa, mediante unos y ceros, la forma de onda analógica correspondiente. Una de las razones para el éxito de la VoIP es la baja relación coste/eficiencia que aporta la posibilidad de transmitir voz codificada ya que somos capaces de representar lo mismo mediante menos bits y por ende reducir el coste debido al menor ancho de banda consumido.

Uno podría pensar que se trata de una relación lineal y que a mismo ancho de banda consumido tenemos la misma calidad. Lo cierto es que esto no es así ya que existen otros factores que afectan a la calidad. Uno de ellos es el esquema de codificación de voz utilizado. Este, cuanto más complejo sea mayor calidad aportará con menos bits. Esto implica algoritmos más complejos que requieren mayor potencia de procesado.

2.3.1. Muestreo de la voz

Para poder crear una representación de una forma de onda analógica (como la voz), necesitamos en primer lugar muestrear la forma de onda y representar casa muestra con un número determinado de bits.

Es obvio que a mayor número de muestras con los mismos bits por muestra más fielmente se representará la forma de onda y por tanto se obtendrá mayor calidad. Lo cierto es que no es necesario ni deseable tomar muestras ilimitadas. El teorema de muestreo de Nyquist nos ayuda en este aspecto. El teorema dictamina que para poder reconstruir una señal es necesario emplear una frecuencia de muestreo como mínimo dos veces la frecuencia máxima de la señal.

Generalmente, el rango en frecuencia de la voz humana abarca desde los 300 hasta los 3800 Hz. Asumiendo una frecuencia máxima menor que 4000 Hz y teniendo en cuenta el teorema de muestreo de Nyquist es necesario establecer como mínimo una frecuencia de muestreo de 8000 Hz.

2.3.2. Cuantificación

La cuantificación es el proceso encargado de convertir una sucesión de muestras de amplitud continua en una sucesión de valores discretos. Para poder representar con exactitud el nivel de amplitud continuo de cada muestra harían falta una gran cantidad de bits por muestra lo cual no interesa.

Al contar con un número limitado de bits por muestra sólo podemos representar un número limitado de niveles de amplitud. La diferencia que existe entre el nivel de amplitud de la muestra original al correspondiente nivel más cercano de la muestra cuantificada se llama ruido de cuantificación[11].

Un aspecto importante de la cuantificación es el uso de los bits. Dado que no afecta de igual manera el ruido de cuantificación a un bajo nivel que a un alto nivel[12] no es necesario utilizar el mismo grado de precisión. Es por ello que se usa la llamada cuantificación no uniforme con la cual se utilizan menos bits para cuantificación de mayores niveles y mayor número de bits para niveles menores.

A continuación se listan diferentes tipos de codecs utilizados para la voz:

2.3.3. G.711

Se trata del códec usado en la PSTN. Dependiendo del lugar donde uno se encuentre se usará uno u otro. En Norte América se utiliza la Ley μ y la Ley a en el resto del mundo. Su tasa binaria es igual a 64 Kbps (una palabra de 8 bits 8000 veces por segundo). Este códec es la fuente de todos los demás.

[11] Si utilizamos 3 bits por muestra podremos representar tan sólo 8 niveles de amplitud. Si el nivel de la muestra antes de cuantificar es de 6.4 nosotros podemos representar 6 con 3 bits pero no 6.4.

[12] La proporción de ruido de cuantificación respecto al nivel de amplitud es mucho mayor cuanto menor es el nivel de amplitud.

2.3.4. G.726

Conocido como ADPCM (*Adaptive Differential Pulse-Code Modulation*), este códec puede funcionar a diferentes tasas binarias (16 Kbps, 24 Kbps, 32 Kbps). Su virtud reside en ofrecer la misma calidad que el G.711 consumiendo la mitad de ancho de banda (en vez de mandar el resultado de la cuantificación, manda la diferencia entre la muestra actual y la anterior). Otra de sus ventajas es el poco trabajo de computación que requiere por parte del sistema.

2.3.5. G. 729A

Mediante el uso de CS-ACELP (*Conjugate-Structure Algebraic-Code-Excited Linear Prediction*), este códec tiene la capacidad de alcanzar una tasa binaria de 8 Kbps ofreciendo una calidad muy buena considerando el ancho de banda que utiliza. El único inconveniente es que es necesario pagar ya que se trata de un códec patentado. Otro de los puntos desfavorables es la alta capacidad de procesamiento que requiere su algoritmia .

2.3.6. GSM

Este códec, Global System for Mobile Communications (GSM), tiene una relación calidad-carga en CPU realmente buena. Consume un ancho de banda de 13 Kbps.

2.3.7. Fail2Ban

Un requisito imprescindible a la hora de incorporar un sistema en producción es que este sea seguro. Al igual que un coche dispone de cerradura y control de alarma, cualquier dispositivo, aplicación o sistema, conectado a la red, debe contener dentro de lo posible los ataques o intentos de intrusión que seguro se darán.

Las distribuciones de Linux cuentan con un conjunto de utilidades destinadas al filtrado de paquetes (*Netfilter*) que proporciona un control de tráfico y seguridad. La aplicación *iptables* permite al usuario configurar las tablas del cortafuegos proporcionado por *Netfilter* .

Fail2Ban es un programa que escanea archivos de registro (*log files*) y *banea* IPs que muestren signos maliciosos (demasiados intentos fallidos de

contraseña, intentos de autenticación con usuario erróneo, etc.). Este actualiza automáticamente las reglas del firewall (*iptables*) de tal manera que el tráfico proveniente de las IPs sospechosas es rechazado. También pueden tomarse otras medidas como enviar un correo electrónico.

A la hora de configurar Fail2Ban es necesario definir **filtros**. Un filtro es una expresión regular que debe corresponder con un patrón de intento de intrusión o cualquier otra expresión (patrones que serán buscados en los logs definidos). Estos filtros son definidos en el directorio ***filter.d***, que se encuentra dentro de la carpeta donde se albergan todos los archivos y directorios de Fail2Ban (*/etc/fail2ban/*). Cada filtro debe ser definido en un archivo de tipo *.conf* con el nombre de la aplicación que vaya a ser protegida, por ejemplo, *asterisk.conf*.

El siguiente archivo importante es ***jail.conf***. Es aquí donde se declaran los *jails*[13]. Los *jails* son definidos mediante la introducción de una serie de parámetros (entre ellos el .conf correspondiente) encabezados por un nombre de carácter descriptivo. Por ejemplo:

```
[asterisk-iptables]

enabled  = true
filter   = asterisk
action   = iptables-allports[name=ASTERISK, protocol=all]
           sendmail-whois[name=ASTERISK, dest=root@localhost,
           sender=fail2ban@pbx.dyndns.org]

logpath  = /var/log/asterisk/messages
maxretry = 5
bantime = 1800
```

donde,

`enable=true` habilita el jail

`filter=asterisk` especifica el filtro a seguir
`action=iptables-allports`... determina la acción o acciones a ejecutar. Estas acciones están definidas en el directorio */etc/fail2ban/action.d* mediante los archivos nombre_de_acción.conf

`logpath=`... el log donde se buscan los patrones definidos en el filtro

[13] Un jail es la combinación de un filtro y una o varias acciones que han de ejecutarse en caso de que se detecte una patrón que se corresponda con las expresiones definidas en el filtro.

maxretry=5 número consecutivo de veces que debe haber una coincidencia de patrón para que se ejecuten las acciones

bantime=1800 tiempo que dura la acción

2.4. Ingeniería de teletráfico

La ingeniería de teletráfico o ingeniería de tráfico, se basa en la aplicación de la teoría probabilística para la solución de problemas relacionados con la planificación, evaluación del rendimiento, operación y mantenimiento de sistemas de telecomunicación[14].

Según la Unión Internacional de las Telecomunicaciones (ITU), el objetivo del teletráfico es formulado como:

Hacer el tráfico medible en unidades bien definidas a través de modelos matemáticos y obtener la relación entre grado de servicio y capacidad del sistema de manera que la teoría se convierte en una herramienta a través de la cuál se pueden basar las inversiones.

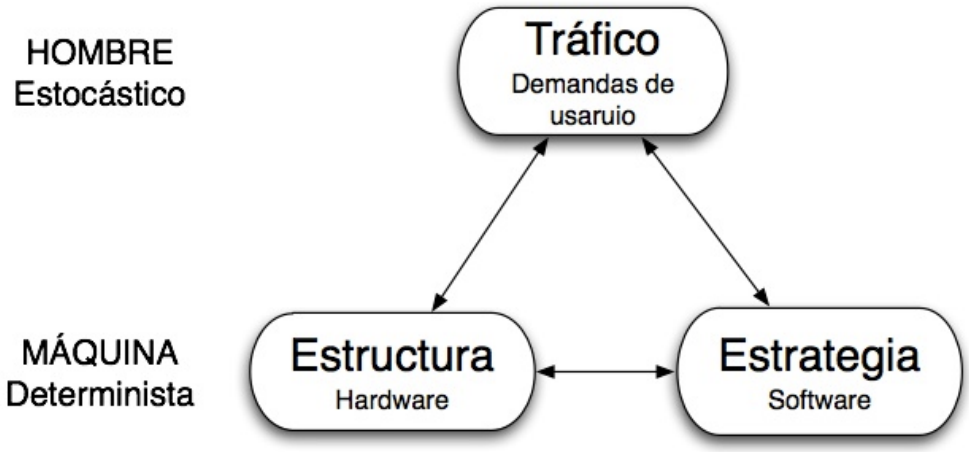


Gráfico 7: Sistema máquina hombre

En el gráfico anterior se muestra la relación hombre/máquina que tiene un sistema de telecomunicaciones. El objetivo final del teletráfico es configurar

[14] La ingeniería de tráfico es aplicable a otras áreas como el tráfico en carreteras o el tráfico aéreo.

sistemas óptimos partiendo del conocimiento de los requerimientos y hábitos de los usuarios.

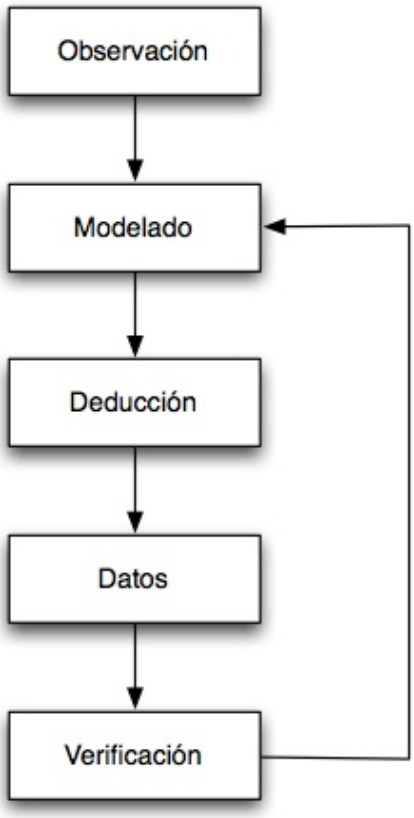


Gráfico 8: Proceso iterativo de validación del modelo teórico.

El Gráfico 7 muestra los pasos que hay que seguir para la validación de un modelo teórico. Este modelo debe ser obtenido a partir de la observación de sistemas reales. Una vez modelado es posible extraer parámetros los cuales pueden ser comparados con las correspondientes observaciones del sistema real. En caso de que se correspondan, el modelo es validado. Si no habrá que modificar y optimizar el modelo.

2.4.7. Conceptos de tráfico

A la hora de planificar un sistema de telecomunicaciones, uno de los objetivos es ajustar los recursos de tal manera que el sistema se ajuste a las variaciones en la demanda del servicio al mismo tiempo que el coste de las instalaciones es mínimo. O dicho de otra manera, el sistema debe ser eficiente.

Para poder cumplir con los objetivos es necesario hacer uso de herramientas y conceptos teóricos. A continuación se explica algunos de ellos.

Concepto de tráfico y la unidad de tráfico "erlang"

Intensidad de tráfico: El tráfico instantáneo en un conjunto de recursos (servidores, trunks, etc.) es el número de recursos ocupados en un instante de tiempo dado. La unidad de medida es Erlang y es adimensional.

Tráfico cursado A_c: Es el tráfico cursado por el conjunto de servidores durante un intervalo de tiempo T. Normalmente, por motivos de dimensionamiento, se usa la intensidad de tráfico media durante un periodo T.

Tráfico ofrecido A: Es el tráfico que sería cursado si no hubiera llamadas rechazadas debido a la falta de capacidad (número de servidores ilimitado). Se trata de un valor teórico que no puede ser medido. Sólo puede ser obtenido a partir del tráfico cursado:

$$A = \lambda \cdot s$$

donde,

λ: número medio de llamadas ofrecidas por unidad de tiempo
s: tiempo medio de servicio

Hora cargada: El tráfico más alto no tiene lugar en la misma franja de tiempo todos los días. La hora cargada se define como los 60 minutos (determinados con una precisión de 15 minutos) los cuales a durante un largo periodo de tiempo tienen, en media, el tráfico más alto.

A la hora de analizar los costes de un sistema es necesario hacer previsiones del comportamiento de este. En la recomendación de la ITU E.506 (*Forecasting international traffic*) se provee de un método para traducir minutos a tráfico en Erlangs el cual nos será de gran utilidad.

En esta recomendación se distinguen dos tipos de estrategia para realizar la previsión: estrategia directa y estrategia compuesta. En ambos procesos, el primer paso es la colecta de datos. Estos datos son la base para crear una previsión del tráfico.

La **estrategia directa** se basa en las medidas de uso del sistema para realizar las conversiones. Por otro lado, la **estrategia compuesta** hace uso de los minutos registrados mensuales para obtener el tráfico y poder así realizar la previsión. Nosotros centraremos el foco en la estrategia compuesta.

Estrategia compuesta

$$A = \frac{Mdh}{60e}$$

donde,

A: tráfico medio estimado en la hora cargada
M: minutos mensuales pagados
d: ratio día-mes
h: ratio hora cargada-día
e: factor de eficiencia

Ratio día mes: Este ratio está relacionado con la cantidad de tráfico cursado en un día laborable típico comparado con la cantidad total de tráfico cursado en un mes. Valores típicos son 0.03-0.04.

Ratio hora cargada-día: cantidad relativa de tráfico medio durante un día laborable en la hora cargada. Este porcentaje puede variar dependiendo de la diferencia horaria entre origen y destino. En la Recomendación E.523 "*Standard traffic profiles for international traffic streams* " se proporciona diferentes distribuciones dependiendo de la diferencia horaria. Un valor alrededor del 10% puede corresponder al valor típico.

Factor de eficiencia: Este factor tiene en cuenta la diferencia entre el tiempo pagado (el tiempo cobrado por el proveedor por tiempo de llamada) en la hora cargada y el tiempo ocupado (el tiempo que se ocupa el circuito) en la hora cargada. Convierte el tiempo pagado en una medida del tiempo total de ocupación del circuito. Este valor es cercano a uno[15].

[15] En la Recomendación E.506, se estima que este factor de eficiencia es un 0.9 aproximadamente. Desde la fecha de publicación de la Recomendación (1992) hasta nuestros días, las velocidades de conmutación han aumentado significativamente y probablemente es posible aproximar el factor de eficiencia a uno.

2.6. *Cloud computing*

Se podría definir cloud computing como un servicio que otorga capacidad informática según-demanda a un conjunto de clientes que para su uso requieren un dispositivo capaz de conectarse a la Red (ordenadores de sobremesa, portátiles, teléfonos inteligentes, tabletas).

Entre sus características podemos citar algunas importantes:

- ✓ Agilidad a la hora de re-provisionar al cliente con nuevos recursos (más capacidad de procesamiento por ejemplo)

- ✓ Reducción de costes gracias a la supresión de barreras de entrada, ya que los costes derivados de este servicio son operacionales ("pagas lo que consumes") y no capitales (no es necesario hacer un gran desembolso inicial para comenzar).

- ✓ Facilidad de acceso: es posible acceder al sistema albergado en la nube desde cualquier buscador en cualquier lugar.

3. Requisitos

Este proyecto trata de cubrir una necesidad real. En nuestro caso se trata de proporcionar los medios necesarios para que la interacción a través de las comunicaciones telefónicas desarrollada por el personal de la empresa sea posible a través de una red VoIP.

El requisito fundamental es el desarrollo y puesta en marcha de una plataforma de pruebas conceptuales que ayude a plantear el análisis de diferentes escenarios de implantación total. Este análisis de diferentes formas de implantación de la plataforma integral de comunicaciones VoIP pretende ayudar a la toma de decisión sobre cuál es la mejor opción.

3.3. Funcionalidad básica

Dado el carácter de investigación del proyecto se trabajará de manera gradual a fin de conocer su rendimiento y respuesta para un posterior estudio sobre el mejor método de implantación.

Teleoperadores

Los TO deberán poder recibir llamadas de clientes y estas a su vez poder ser traspasadas a técnicos en servicio (TPN, TSN). Al mismo tiempo deben tener la opción de dejar la llamada en espera.

Igualmente, la recepción de llamadas debe estar provista de un sistema de colas capaz de distribuir eficientemente cada llamada.

Técnicos

Además de las funcionalidades básicas para recepción de llamadas, tanto los TPN como los TSN deben tener la posibilidad de realizar llamadas a cualquier cliente independientemente de su localización (llamada a línea fija, llamada a red móvil o llamada internacional).

Registro y grabación de llamadas

El sistema debe dar la posibilidad de realizar un seguimiento de las llamadas efectuadas y recibidas así como la consulta de los archivos de grabación correspondientes a estas llamadas.

Seguridad

El entorno debe estar provisto de herramientas destinadas a obtener un sistema seguro contra ataques o intentos falsos de identificación.

IV. DISEÑO Y SOLUCIÓN TÉCNICA

1. Introducción

Durante esta sección se expondrá primeramente la solución implantada para la plataforma de pruebas conceptuales. Seguidamente se proseguirá con el análisis de cada una de sus partes y sus interconexiones. Se proseguirá con la descripción del diseño así como su configuración. Finalmente se dispondrá del estudio de ventajas entre varios escenarios de implantación así como sus costes.

2. Plataforma básica de pruebas

2.1. Arquitectura

A continuación se presenta un gráfico que muestra la arquitectura general del sistema. En ella podemos distinguir varias partes diferenciadas. El corazón de la plataforma se encuentra en las instalaciones de Actualize S.L. en Alcobendas. Se trata de un servidor Dell PowerEdge R6 albergado en red local . Este cuenta con sistema operativo Linux CentOS 5.5 instalado mediante el todo-en-uno Asterisk Now 1.7 que se proporciona en la página de Asterisk[16]. Para que los técnicos y tele-operadores pudieran llamar desde el ordenador hemos utilizado el softphone gratuito X-lite de la empresa Counterpath.

Por otro lado hemos contratado un proveedor de VoIP que nos ha provisto de dos números virtuales DID para poder emitir y recibir llamadas a través de ellos.

[16] http://www.asterisk.org/downloads

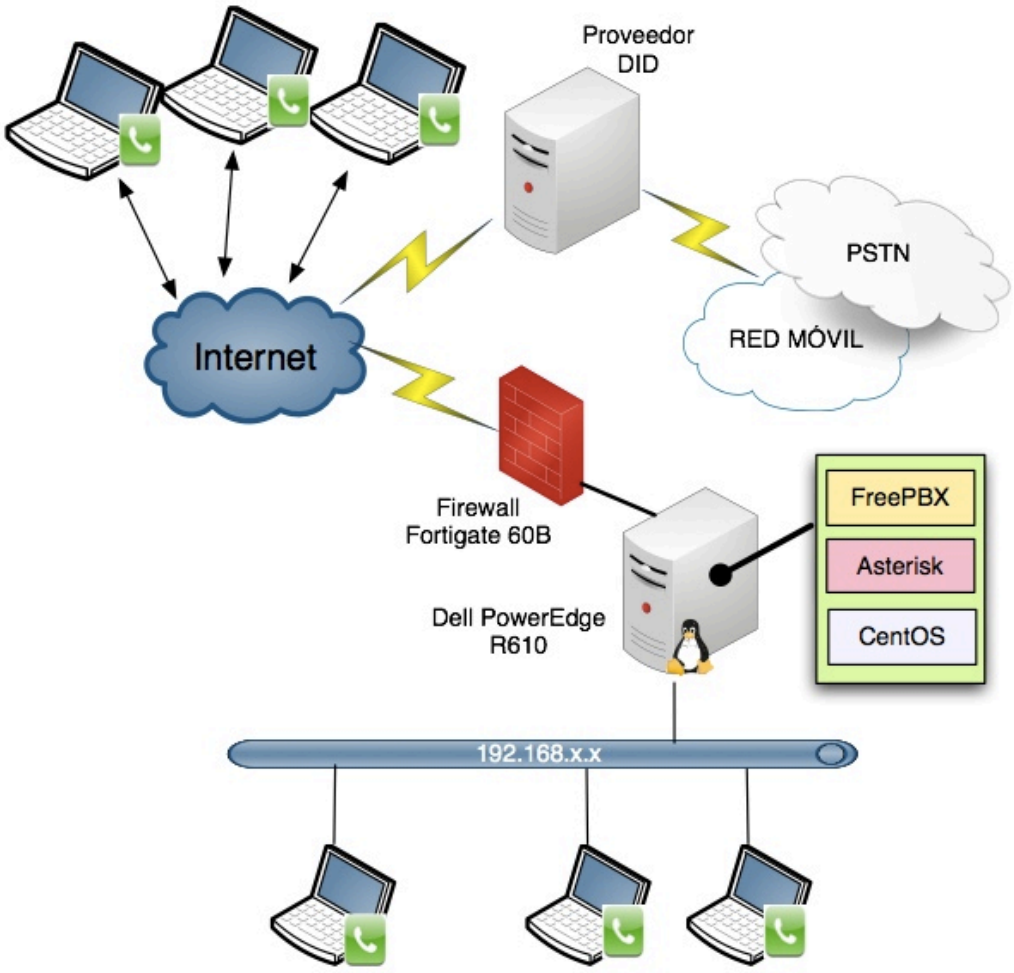


Gráfico 9: Arquitectura plataforma básica VoIP

2.2. Servidor

El servidor que ha sido utilizado para albergar Asterisk y los demás programas que aportan funcionalidad a la plataforma ha sido un Dell PowerEdge R610. Este aporta con creces las necesidades de capacidad de procesado y memoria que son necesarias para nuestra plataforma.

2.3. Softphones

Un softphone es un programa que realiza la función de teléfono principalmente para llamar a otros softphones o dispositivos que ”hablen” el mismo protocolo (SIP).

Existe gran variedad de softphones, siendo muchos de ellos gratuitos. Tras probar varios de ellos elegimos X-Lite de la empresa CounterPath por varias razones:

- ✓ Facilidad de uso.
- ✓ Soporte de protocolo utilizado por Asterisk (SIP).
- ✓ Popularidad. Gran aceptación entre usuarios de la comunidad Asterisk.

Gráfico 10: X-Lite softphone

2.4. Proveedor VoIP

Un proveedor de VoIP proporciona diferentes servicios para comunicaciones telefónicas a través de la Red. Entre ellos, probablemente el que más relevancia tiene es la capacidad de proporcionar un enlace con la PSTN de manera rápida y sencilla. Rápida porque la asignación de un DID (*Direct Inward Dialing)* no suele sobrepasar unos minutos y sencilla por la fácil configuración por nuestra parte para poder realizar llamadas a través de su puerta de enlace.

Nuestros requerimientos para la plataforma de pruebas conceptuales en este ámbito son:

- 1 DID con el que poder emitir llamadas

- 1 DID con el que poder recibir llamadas

Un DID es una característica ofrecida por las compañías telefónicas (controladoras de los números de teléfono) por la cual se asocia un número telefónico con un *trunk* conectado a la PBX de la empresa que haya contratado el servicio. En un entorno VoIP, siendo el concepto el mismo, es el proveedor IP quien media en el origen y terminación[17] de llamadas en PSTN.

La decisión de elegir un DID de entrada y otro de salida viene propiciada por política de empresa con sus clientes ya que, por contrato, es necesario que los clientes finales visualicen un número de teléfono cuando se les llama. Si utilizáramos un solo DID (un solo número) para ambas llamadas, salientes y entrantes, el cliente podría volver a llamar al teléfono que visualiza en la pantalla de (por ejemplo) su teléfono móvil. Esto ocasionaría problemas ya que si el cliente final llama este teléfono no será atendido por el departamento con quien haya hablado, sino con el que haya sido configurado para la recepción de llamadas.

La empresa elegida para esto ha sido Affinalia, el actual proveedor de canal primario de Actualize. Serán ellos los encargados de proporcionarnos los detalles de conexión que deben ser configurados en Asterisk.

2.5. Diseño y configuración de la plataforma de pruebas conceptuales

En este apartado se expondrá el diseño de cada uno de los componentes de la solución para la plataforma de pruebas conceptuales así como los pasos seguidos para su configuración y posterior puesta en funcionamiento.

2.5.1. Diseño

Extensiones

En el punto 2.1.2.3. (III. PLANTEAMIENTO DEL PROBLEMA) se explicó el concepto de extensión dejando claro que una extensión en Asterisk puede ser o no el número asociado a una persona. FreePBX hace uso de su primera acepción ya que basa el conjunto de sus aplicaciones en torno al concepto de

[17] Del inglés, *origination* y *termination,* significa que las llamadas se originan en la PSTN (hacia otra red como es la red VoIP) o terminan en ella.

extensión==persona asociada. Esta aproximación es natural ya que la mayor parte de las empresas interesadas en el uso de un programa como Asterisk lo eligen basadas en la necesidad de emular una PBX.

Con la intención de tener un sistema escalable y una buena organización se ha diseñado un plan de extensiones de acuerdo a la naturaleza y características de la empresa.

Al tratarse de una multinacional debemos reservar un dígito para distinguir entre las diferentes delegaciones de cada país. Por otro lado, dado que cada delegación cuenta con diferentes departamentos, es necesario reservar otro dígito para diferenciarlos unos de otros. En último lugar debe poder identificarse el usuario al que se quiere llamar; para ello hemos reservado dos dígitos. De esta manera, si marcamos la extensión 1236 se estará llamando al usuario 36 del departamento 2 del país 1. A continuación se muestra una tabla con el esquema anteriormente explicado y el listado de extensiones:

PLAN DE EXTENSIONES			
Formato: ABCD			
	A: Delegación		
	B: Departamento		
	CD: Usuario		
Asignación de numeración			
	Delegaciones		
		Madrid	1
		Bogotá	2
		México	3
		Sº de Chile	4
		Barcelona	5
		Londres	6
		Sao Paulo	7
	Departamentos		
		Dirección	1
		Admin.	2
		HR	3
		Technology	4
		Marketing	5
		Quality of Service	6
		Comercial	7
		ISC	8
		Operations	9

Tabla 1: Plan de extensiones

Técnicos 2º N.		
Roberto	1406	Contraseña1
Alexandre	1407	Contraseña2
Carlos	1408	Contraseña3

Tele-operadores		
Nuno	1480	Contraseña4
Lorena	1481	Contraseña5
Susi	1482	Contraseña6

Tabla 2: Distribución de extensiones

El siguiente paso consistirá en instalar y configurar los softphones en cada uno de los ordenadores de los técnicos y tele-operadores. Para proceder de una manera eficaz y organizada se redactó un tutorial dedicado a este propósito[18]

Trunks (DIDs)

Dado que nuestras necesidades para esta plataforma de pruebas son mínimas (sólo dos DIDs) un solo proveedor de DIDs es suficiente. Esto es importante porque los precios de un proveedor a otro pueden variar dependiendo de su localización geográfica. Existen proveedores internacionales de DIDs pero estos suelen ser más costosos que un proveedor local o nacional.

Los números proporcionados por Affinalia son: **91xxx1801** y **91xxx1802**[19]. Utilizaremos el primero, 91xxx1801, para emisión de llamadas y el segundo para recepción.

Es costumbre en algunas organizaciones marcar el 0 anterior al número a llamar. Esto tiene sentido en una PBX tradicional dado que es una forma de indicar a la PBX que se llama al exterior de tal manera que tan pronto se marque el 0 la central conecta con el canal reservado a llamadas externas. Esto no es necesario hacerlo cuando se utiliza Asterisk como PBX ya que el número marcado es transmitido, no de manera secuencial como en un teléfono tradicional (DTMF), sino todo "de golpe" a través de una petición INVITE del

[18] Apéndice B: tutorial X-Lite

[19] Los números han sido parcialmente omitidos por política de la empresa.

protocolo SIP. Después es Asterisk quien interpreta el número y dirige la llamada de acuerdo al contexto al que pertenezca el usuario[20] y la extensión[21] marcada.

Rutas de entrada

Estableceremos una ruta de entrada para el DID elegido para recepción de llamadas. Esta ruta desembocará en un una cola que llamará de manera equitativa a los tele-operadores alistados utilizando para ello planificación *round-robin*.

Ruta de salida

Se deberá configurar una ruta de salida de manera que las llamadas de los técnicos al exterior sean dirigidas a través de esta ruta. Tienen que tener la posibilidad de hacer llamadas locales/nacionales, a teléfonos móviles y a números internacionales.

Buzón de voz

En caso de que no haya tele-operadores disponibles la llamada debe ser transferida a un buzón de voz. Previamente, debe reproducirse un mensaje indicando lo siguiente: "*Su llamada no puede ser atendida en estos momentos. Por favor, deje su mensaje indicando su número de teléfono y nos pondremos en contacto con usted tan pronto como sea posible. Gracias.*" Los técnicos tendrán acceso a este buzón y podrán consultarlo para devolver las llamadas.

2.5.2. Configuración

2.5.2.1. Creación extensiones

Técnicos

Para crear una extensión debemos acceder a ***extensions*** dentro de la sección ***applications***. A continuación se listan los campos rellenados para la extensión 1406 perteneciente a Roberto (de manera similar configuramos las extensiones de los otros dos técnicos, Alexandre y Carlos). Antes notar que el correo electrónico configurado para el envío de mensajes del buzón de voz será al que todos los técnicos accedan para consultar los mensajes:

[20] Hay que recordar que aunque FreePBX no "obligue" al administrador a definir usuarios y contextos, FreePBX reorganiza la información introducida por su interfaz y la traduce en sintaxis válida para Asterisk, creando usuarios, contextos y una lógica de dialplan.

[21] Recalcar la interpretación por parte de Asterisk del concepto extensión

Add Extension

- *User Extensión*: 1406
- *Display Name*: Roberto

Device Options

- *secret*: Contraseña1
- *nat*: never
- *Lenguage Code*: es

Recording Options

- *Inbound External Calls*: Always
- *Outbound External Calls*: Always
- *Inbound Internal Calls*: Always
- *Outbound Internal Calls*: Always

Voicemail

- *Voicemail Password*: Contraseña1
- *Email Address*: correo@técnicos.com
- *Email Attachment*: Yes
- *Play Envelope*: Yes
- *Delete Voicemail*: Yes

Teleo-operadores

La configuración de extensiones para los tele-operadores será la misma excepto el buzón de voz, que no es necesario configurarlo.

2.5.2.2. Configuración del trunk

Para la configuración del trunk fue necesario contactar con nuestro proveedor, Affinalia, para que nos proporcionara algunos detalles relativos al registro.

Sobre la barra de menú posterior, pinchamos en *Connectivity>Trunks*. Seguidamente, en la parte posterior derecha de la pantalla, pinchamos en *Add Trunk*. Dado que nuestro proveedor lo que nos proporciona es un trunk SIP es esta la opción que debemos seleccionar.

- *Trunk Name:* Affinalia
- Outbound CallerID: <usuario>
- *Trunk Name: Affinalia*
- *PEER Details:*
    - host=dominio.net
    - insecure=invite
    - username=usuario
    - secret=contraseña
    - type=peer
- *Register String:* usuario@nuestraIP:comntraseña:usuario@dominio.net

El usuario, contraseña y dominio.net son detalles proporcionados por Affinalia.

Antes de configurar las rutas de entrada y salida es necesario configurar la cola a la cual serán destinados aquellos clientes que llamen al número para recepción de llamadas.

2.5.2.3. Configuración de la cola

Pinchando sobre el panel posterior en ***Applications>Queues*** nos situamos en la pantalla principal de adición de colas. Los detalles son:

- *Queue Number*: 222
- *Queue Name*: ColaTele
- *Queue Password*: Contraseñacola
- *Static Agents*:

    1480
    1481
    1482
- *Ring Strategy*: rrmemory (Round Robin Memory)
- *Call recording*: wav
- *Recording Mode*: After Answered
- *Join Empty*: strict (se acepta al llamante en la cola siempre y cuando sea posible atenderle (hay agentes listados en la cola))
- *Fail Over Destination*: *Announcement*: MensajeBuzon (este "*Announcement*" será creado en el siguiente punto)

2.5.2.4. Configuración rutas de entrada y salida

Ruta de entrada

Para la ruta de entrada es necesario que pinchemos sobre ***Connectivity>Inbound Routes***. Pinchamos sobre ***Add Incoming Route***. Los parámetros necesarios para la configuración de la ruta son los siguientes:

- *Description*: Recepción_llamadas
- *DID Number*: 91xxx1802
- *Lenguage*: es
- *Set Destination*: Queues > ColaTele

Ruta de salida

En este caso pinchamos en ***Outbound Routes***.

- *Route Name*: salida
- *Dial Patterns that will use this Route*:

prepend	prefix	match pattern	CallerID
-	-	[968]XXXXXXXX	-
-	-	0.	-

- ✓ **Prepend:** dígitos a añadir en caso de que exista coincidencia con match pattern
- ✓ **Prefix:** prefijo que será suprimido si existe coincidencia con match pattern
- ✓ **Match pattern:** el número marcado será comparado con el *prefix+match pattern*. En caso de que exista coincidencia sólo se enviará al *trunk* el *match pattern*[22].

Grabación y configuración de mensaje

El primer paso que hay que tomar es realizar la grabación del mensaje. Esto puede hacerse de dos formas. Una grabando el mensaje directamente desde el softphone o subiendo un archivo .wav. Nosotros lo haremos mediante la grabación directa.

Para poder salvar y cargar la grabación que realicemos mediante el uso del softpohone hay que:

[22] En el apéndice D: *pattern-matching* se puede consultar la sintaxis

1. Dirigirnos a ***Admin>System Recordings***.
2. Introducir la extensión del dispositivo desde el que queramos grabar el audio.
3. Marcar el código[23] *77 y esperar a la señal para la grabación del audio.
4. Marcar # una vez hayamos terminado con la grabación (en caso de querer escuchar el audio grabado basta con marcar *99. Si lo que queremos es repetir el proceso hay que volver a marcar *77)
5. Esperar a la actualización de la página de grabaciones de FreePBX.
6. Nombrar el audio grabado.
7. Salvar.

Una vez tenemos el audio ya podemos proceder a crear una notificación (***Announcement)***. Para ello pinchamos, en el panel principal, en ***Applications>Announcements***:

- *Description*: MensajeBuzón
- *Recording*: MensajeBuzón
- *Destination after playback*: Voicemail : <1406> Roberto

El último paso consiste en configurar X-Lite para cada usuario. Esto puede ser revisado en el tutorial que se incluye en el apéndice tal como se indicó anteriormente.

2.5.2.5. Configuración del Fortigate 60B y dispositivos para traspaso SIP y RTP

Los puertos UDP con los que trabaja el protocolo SIP y RTP son el 5060 y los especificados en el archivo de configuración de Asterisk /etc/asterisk/*rtp.conf* (10001-20000). Para que Asterisk reciba el tráfico destinado a estos puertos es necesario redirigir estos puertos al servidor que alberga Asterisk (IP 192.168.1.204).

Por otro lado es necesario especificar los puertos que serán usados para el tráfico RTP no sólo en el archivo *rtp.conf* de Asterisk, sino también en los softphones X-lite.

[23] Asterisk tiene reservados unos códigos para funciones especiales.

2.6. Seguridad: Fail2Ban

En primer lugar, dado que Fail2Ban está escrito en Python es necesario instalar los paquetes correspondientes del lenguaje. Utilizamos yum[24] para la instalación:

```
yum install python
yum install fail2ban
```

A continuación creamos el filtro dedicado para asterisk:

```
cd /etc/fail2ban/filter.d
nano asterisk.conf
```

Introducimos las expresiones regulares convenientes:

```
Failregex = NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Wrong password
            NOTICE.* .*: Registration from '.*' failed for '<HOST>' - No matching peer found
            NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Username/auth name mismatch
            NOTICE.* <HOST> failed to authenticate as '.*'$
            NOTICE.* .*: No registration for peer '.*' (from <HOST>)
            NOTICE.* .*: Host <HOST> failed MD5 authentication for '.*' (.*)
```

El siguiente paso es crear el correspondiente apartado en */etc/fail2ban/jail.conf*:

```
[asterisk-iptables]

enabled  = true
filter   = asterisk
action   = iptables-allports[name=ASTERISK, protocol=all]
           sendmail-whois[name=ASTERISK, dest=correo@uc3m.es, sender=fail2ban@asterisk.com]
logpath  = /var/log/asterisk/messages
maxretry = 5
bantime = 1800
```

Notar que una de las acciones es mandar un correo de tal manera que se nos avisará cuando se haya producido un bloqueo (las direcciones no se corresponden con direcciones reales).

Por otro lado hay que habilitar el registro de mensajes tipo "*notice*" en el archivo de log de asterisk:

```
nano /etc/asterisk/logger.conf
```

[24] *Yellow Dog Updater* (YUM) es una herramienta libre de gestión de paquetes para sistemas Linux basados en RPM (Red Hat Packet Manager)

messages => notice,debug

Por último indicamos que el servicio sea habilitado (para todos los niveles de ejecución) mediante el comando **chkconfig** y a continuación iniciamos el servicio:

```
chkconfig fail2ban on
/etc/init.d/fail2ban start
Starting fail2ban:         [  OK  ]
```

2.7. Análisis diferentes implantaciones

Con la propuesta de plataforma en pruebas de VoIP, se ofrece a la empresa una primera solución a la migración gradual de su sistema de comunicaciones tradicional. Esta primera solución provee un sistema alternativo al actual con el que poder empezar a ahorrar costes de inmediato[25] y dar un paso más en la migración al sistema definitivo.

[25] Las llamadas a través de DIDs son más baratas que el actual medio de emisión de llamadas que es Skype.

Destino	Skype (€/hora)		Vozelia (€/hora)		% ahorro
España (fijo)	0,019	1,14 €	0,015	0,90 €	21,05%
España (móvil)	0,220	13,20 €	0,049	2,94 €	77,73%
Francia (fijo)	0,019	1,14 €	0,012	0,74 €	35,26%
Francia (móvil)	0,169	10,14 €	0,077	4,62 €	54,44%
Suecia (fijo)	0,019	1,14 €	0,012	0,72 €	36,84%
Suecia (móvil)	0,238	14,28 €	0,050	3,00 €	78,99%
México (fijo)	0,085	5,10 €	0,069	4,14 €	18,82%
México (móvil)	0,267	16,02 €	0,058	3,48 €	78,28%
Uruguay (fijo)	0,105	6,30 €	0,059	3,54 €	43,81%
Uruguay (móvil)	0,215	12,90 €	0,184	11,04 €	14,42%
China (fijo)	0,019	1,14 €	0,013	0,78 €	31,58%
China (móvil)	0,019	1,14 €	0,013	0,78 €	31,58%

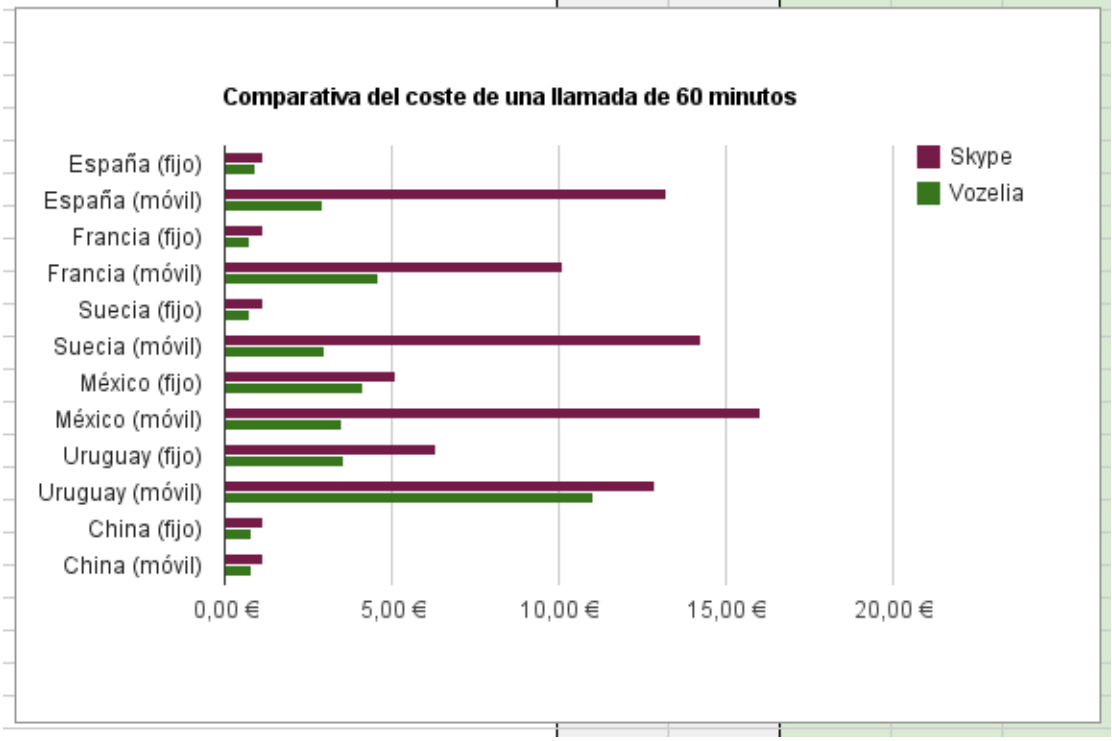


Gráfico 11: Comparativa €/hora Skype vs. Proveedor VoIP. Fuente: SinoLogic

Parte de los requisitos a la hora del planteamiento del problema eran proveer herramientas que sirvan como base para el fundamento en la toma de decisiones sobre el mejor escenario para una implantación global.

Dos son las propuestas. La plataforma puede ser distribuida en todas las delegaciones de los diferentes países o contratar a una empresa que proporcione capacidad informática con tamaño modificable en la nube.

2.7.1. Cloud vs. *On-premises*

Existen diversas empresas que ofrecen servicios en la nube (Amazon Elastic Compute Cloud, Windows Azure entre otras). En todas ellas se sigue el modelo general de "pagar por lo consumido". Esta es una gran ventaja si se quiere comenzar a usar el sistema que vaya a utilizarse en la nube en un tiempo despreciable comparado con el que se empelaría en el caso de utilizar recursos propios.

Hablamos de un modelo *on premises* cuando nuestro sistema se encuentra albergado en nuestras instalaciones. El desembolso inicial necesario para implementar un sistema *on premises* no puede ser despreciado. Sin embargo, implementar el sistema en la nube tiene un desembolso inicial nulo[26] comparado con el de un sistema *on premises.* Es por esto que es necesario comparar ambos escenarios y analizar sus costes en función del volumen de producción (número de minutos de llamadas entrantes y salientes del sistema).

Una vez se disponga de los costes derivado de cada uno de los escenarios en función de los minutos llamados, podremos determinar, en función de las previsiones futuras, qué escenario es el óptimo.

Para el estudio económico hemos tenido en cuenta diversos factores para cada uno de los escenarios propuestos:

Cloud

- *Sistema Operativo:* si optamos por cualquier distribución Linux (exceptuando RedHat) el coste será cero euros. Si por el contrario se elige Windows esto incurrirá en gasto.
- *Ancho de banda (BW)*: las dos empresas consultadas han sido Azure y Rackspace. Ambas cobran por ancho de banda acumulado. Es decir, la cantidad de bits consumidos por la interfaz de red a lo largo del mes. Para esto sólo es necesario tener en cuenta el tráfico de salida ya que el consumo de datos entrantes en la nube de Azure (al igual que en Rackspace) no conlleva gasto alguno.
- *Memoria RAM*: Según la capacidad de Memoria RAM que requiramos ofrecen unos precios u otros. Cuanto más RAM mayor coste.

[26] La tarificación que emplean las empresas dedicadas al *cloud computing* se basa en un modelo de cobro por recurso utilizado mensualmente. De esta manera debería tener que pagarse hasta que venza un periodo de facturación.

On Premises

- *Servidor dedicado*: será necesario adquirir un servidor para cada una de las sedes principales.
- *Ancho de banda*: en *cloud* se paga por el ancho de banda consumido mientras que teniendo los servidores en nuestras instalaciones hay que contratar un enlace de acceso dedicado y simétrico.

Por otro lado tenemos un factor común, los DIDs contratados.

Gracias a los ACDs[27] que la empresa tiene en España y en Latinoamérica tenemos cifras de los minutos entrantes y salientes en cada delegación. Las variables que juegan en el resultado que se mostrará a continuación son:

Distribución de llamadas

Número minutos llamadas entrantes, salientes y minutos en Hora Cargada en:

- España
- Colombia
- México

Los minutos en Hora Cargada han sido calculados mediante la siguiente fórmula:

$$MinHC = Mdh$$

donde,

M: Minutos totales en un mes
d: ratio día-mes
h: ratio hora cargada-día

Para el cálculo del ancho de banda requerido es necesario saber el número de llamadas concurrentes que tendrán lugar en el sistema. El cálculo del número de llamadas concurrentes se ha calculado de la siguiente manera:

Para llamadas entrantes y salientes: $MLLC = \frac{MinHC}{tmll}$

[27] *Automatic Call Distributor*

Donde,

MLLC: Máximo número de llamadas concurrentes
tmll: tiempo medio de llamada = 8 min

Al tener el número máximo de llamadas concurrentes podemos calcular el ancho de banda necesario en los enlaces dedicados (*on premises*). Para el cálculo del ancho de banda en *cloud* no es necesario el número máximo de llamadas concurrentes. Los motivos se explicaron anteriormente.

Quedaría por calcular el gasto derivado de la contratación de DIDs. Al necesitar DIDs con números pertenecientes a diferentes países ha sido necesario buscar varios proveedores para encontrar aquellos que no cobran por la recepción de llamadas. Estos proveedores son *mydivert.com* para el caso de Colombia, *didlogic* para España, y *voipms* para México. En total Actualize tiene 104 DIDs.

El cálculo del gasto por la emisión de llamadas (€/min) se ha calculado en varios pasos:

1. Se estima que un 80% de llamadas se hacen a teléfonos fijos y un 20% a teléfonos móviles
2. Teniendo la información anterior y consultando las tarifas de los diferentes proveedores es posible calcular los precios de emisión de llamadas desde un país a las diferentes delegaciones.

Para todos los cálculos sólo se han considerado la utilización del códec G.711 siendo este el caso más desfavorable (mayor ancho de banda consumido). Por otro lado cabe decir que los costes son orientativos y están sujetos a cambio una vez se consulten algunas de las tarifas reales como las relativas a los enlaces dedicados.

Las siguientes tabla muestran esta información:

COSTE LLAMADAS EMITIDAS DESDE (cent. €/minuto)

Colombia				
mydivert.com				
Destino				
SP		cent/min	%	cent/min*%
		1,04	80	0,832
	Mobile	5,04	20	1,008
	Tot.			**1,84**
CO				
	Fixed	1,44	80	1,152
	Mobile	5,12	20	1,024
	Tot.			**2,176**
MX				
	Fixed	1,12	80	0,896
	Mobile	5,2	20	1,04
	Tot.			**1,936**

Tabla 2: Coste llamadas emitidas desde Colombia

España				
didlogic				
Destino				
SP		cent/min	%	cent/min*%
	Fixed	0,704	80	0,5632
	Mobile	3,968	20	0,7936
	Tot.			**1,3568**
CO				
	Fixed	1,184	80	0,9472
	Mobile	3,472	20	0,6944
	Tot.			**1,6416**
MX				
	Fixed	2,552	80	2,0416
	Mobile	0,672	20	0,1344
	Tot.			**2,176**

Tabla 3: Coste llamadas emitidas desde España

Mexico				
voipms				
Destino				
SP		cent/min	%	cent/min*%
	Fixed	1,008	80	0,8064
	Mobile	1,096	20	0,2192
	Tot.			**1,0256**
CO				
	Fixed	1,008	80	0,8064
	Mobile	5,592	20	1,1184
	Tot.			**1,9248**
MX				
	Fixed	1,624	80	1,2992
	Mobile	9,64	20	1,928
	Tot.			**3,2272**

Tabla 4: Coste llamadas emitidas desde México

Con toda esta información ya es posible calcular los gastos totales en base al número de minutos consumido por mes.

Cloud			
Rackspace		Azure	
CF (€/mes)		CF (€/mes)	
4GB RAM	140	3.5GB RAM	41
DIDs	301	DIDs	301
subtot.	**441**	subtot.	**342**
CV		CV	
BW	10	BW	6
gasto mins	471	gasto mins	471
subtot.	**481**	subtot.	**477**
tot.	**922**	tot.	**819**
Total Cloud	**819**		

Tabla 5: Análisis costes en Cloud

On premises					
España		**Colombia**		**Mexico**	
CF (€/mes)		**CF (€/mes)**		**CF (€/mes)**	
Servidor	20	Servidor	20	Servidor	20
DIDs	125	DIDs	125	DIDs	125
subtot.	**145**	subtot.	**145**	subtot.	**145**
CV		**CV**		**CV**	
BW	206	BW	206	BW	206
gasto mins	189	gasto mins	97	gasto mins	153
subtot.	**395**	subtot.	**303**	subtot.	**359**
tot.	**541**	tot.	**449**	tot.	**504**

Total On premises	**1493**

Tabla 6: Análisis costes *On premises*

Con este modelo se ofrece, tal como se planteó en los objetivos, una herramienta con la que poder evaluar el coste aproximado de la implantación de ambos escenarios.

V. PRESUPUESTO

Descripción	
Titulo	**DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE VOIP BASADO EN ASTERISK Y PABX**
Duración (meses)	4
Tasa de costes indirectos	20%

Presupuesto total del proyecto (€)

€ 3.948,00

Desglose presupuestario (costes directos)

PERSONAL

Apellidos, Nombre	Categoría	Dedicación (hombres mes)	Coste hombre mes	Coste (€)
Leguina, Iñigo	IT Director	0,08	4250	340
Tatay, David	IT Manager	0,1	2100	210
Gª de Vinuesa, Borja	Ingeniero	1,4	1900	2660
			total	3210

1 hombre mes=141.22

EQUIPO

Descripción	Coste(€)	% Uso dedicado Proyecto	Dedicación (meses)	Periodo de depreciación
Serv. PowerEdge R610 con CentOS	1200	100%	4	60
Softphone X-Lite	0	100%	4	60
				total

Otros costes directos del proyecto

Descripción	Empresa	Coste imputable
	total	0

Otros costes directos del proyecto

Descripción	Empresa	Coste imputable
	total	0

Resumen de costes	
Tipo	Prespuesto costes totales
Personal	3210
Amortización	80
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes indirectos	658
Total	3948

VI. RESULTADOS Y EVALUACIÓN

1. Plataforma de pruebas conceptuales

En primera instancia se tuvo que dedicar buena parte del tiempo a depurar el sistema ya que el corta fuegos tiene un comportamiento anómalo frente al tráfico de VoIP y más concretamente frente al protocolo SIP. El problema reside en el intento por parte del corta fuegos de ayudar en el traspaso de flujo SIP. Este "rompía" los paquetes (cambio de puertos) SIP a la hora de redirigirlos al servidor Asterisk. Como consecuencia, el servidor intentaba un número determinado de retransmisiones hasta que, al no obtener respuesta, cortaba la conexión. Se pidió ayuda al centro de soporte de Fortinet pero hasta día de hoy no hemos obtenido una resolución clara del problema.

Se adoptaron medidas alternativas para poder continuar con el desarrollo del proyecto. Las medidas propuestas fueron dos: (1) contratar un enlace dedicado y establecer un corta fuegos de por medio en modo transparente o (2) hacer uso de la VPN ya que en este caso el corta fuegos no interviene los paquetes. Finalmente, como solución provisional, se adoptó esta segunda.

La plataforma se ha mantenido estable durante las cuatro semanas en las que ha estado en funcionamiento. La calidad del audio ha sido buena según las opiniones de los técnicos y tele-operadores que utilizaron la plataforma. Sólo en casos puntuales surgieron problemas debidos a fallos ajenos a la plataforma (problemas derivados de la colisión de IPs entregadas por el DHCP principalmente).

El códec empleado (G.711) proporciona una calidad similar a la de telefonía tradicional (64 Kbps). El ancho de banda consumido por el máximo número de llamadas concurrentes (3) no sobrepasó en ningún momento la capacidad del enlace aun compartiendo ancho de banda con el tráfico de navegación de la empresa.

Cabe destacar el trabajo que conllevó la depuración del sistema. La existencia de múltiples saltos en la plataforma básica hizo que la resolución de cualquier problema resultara algo no demasiado sencillo. Para determinar dónde se encontraban los problemas (por ejemplo, la ruptura de paquetes por parte del corta fuegos) hubo que realizar capturas con diferentes programas[28] y revisar todo el flujo de sesión.

2. Análisis cloud vs. *on premises*

En base a los resultados reflejados en las tablas 5 y 6, observamos que la solución en *cloud* (Azure) es menos costosa que una solución distribuída (*on premises*). A primera vista, debido a este importante factor económico, parece ser la solución con mayor ventaja. Sin embargo existen otros factores no económicos que hay que tener en consideración.

La solución *cloud* tiene una ventaja, y es que su puesta en marcha es casi inmediata. A pesar de ello, es un sistema que hay que analizar a fondo ya que la plataforma se encontraría en un lugar no gestionado por la empresa y, aunque los proveedores del servicio den unas altas garantías de estabilidad, no tenemos un control absoluto sobre nuestro sistema.

[28] Wireshark (versión Windows para los clientes SIP y *tshark* para captura de tráfico en el servidor) y un programa proporcionado por el cortafuegos llamado *sniffer*. Puede consultarse el manual propio realizado en su día para la utilización de estas herramientas (Apéndice C: Documentación capturas)

Tener un sistema *on premises* es más caro, pero posiblemente más seguro. Somos nosotros quienes los gestionamos teniendo un acceso directo a cualquier parte de la arquitectura. Por otro lado, al ser un sistema distribuido es fácil aplicar redundancia. Probablemente también lo sea en el caso de tener el sistema en la nube pero esto hay que estudiarlo.

Durante el diseño del modelador de costes se realizó una calculadora de probabilidad de espera en la cola, usando para ello la función Erlang C. Esta, introduciendo como parámetros el número de agentes y el tráfico en Erlangs, calculaba la probabilidad de espera en cola. El objetivo de esto era introducir herramientas con las que poder calcular los recursos requeridos para un grado de servicio concreto. No ha sido introducida en el proyecto debido a su poca practicidad.

VII. CONCLUSIONES

Con este proyecto se ha podido comprobar que la puesta en marcha de un sistema básico de telefonía IP adaptado a las necesidades de un cliente no tiene por qué acarrear una excesiva complejidad. Sin embargo, si el contexto empresarial impone una implantación más cuidadosa (evaluación de modelo económico y variación según estrategia futura) la complejidad se incrementa.

Durante el desarrollo del proyecto ha existido un proceso de aprendizaje técnico continuo. Manejar con soltura los conceptos clave de Asterisk es clave aun teniendo una interfaz gráfica que haga de lo "difícil" algo sencillo.

Cabe destacar este último punto. Durante el proceso de desarrollo de la plataforma básica existieron dos versiones paralelas. Una con FreePBX y otra sin interfaz gráfica. En mi opinión esta última es la que brinda mayores posibilidades en relación con el tiempo empleado. Esto, intuitivamente, puede parecer una contradicción pero nada más lejos de la realidad. El tener una interfaz gráfica de por medio oculta el diseño interno con el que se crea funcionalidad. Es FreePBX el impositor de la edición y programación de los archivos de configuración de Asterisk. Es cierto que FreePBX tiene unos archivos de configuración específicos para realizar cambios manuales en los archivos de configuración de Asterisk. Sin embargo, la estructura base del código[29] es realizada por FreePBX haciendo el sistema poco transparente.

En nuestro caso se optó por implementar FreePBX por la rapidez de resultados, usabilidad y su gran interfaz de reportes a través de CDR dado que se trataba de un entorno de pruebas.

En mi opinión, el diseño y configuración de la plataforma sin intermediarios debería resultar en un mayor control del sistema y una ganancia en flexibilidad y funcionalidad. Sin duda es un campo en el que me gustaría seguir trabajando ya que las posibilidades que brinda son realmente grandes.

[29] Me refiero al código que Asterisk tiene para la edición de sus archivos de configuración y no el código con el que los módulos están implementados (lenguaje C)

Resumiendo, creo que este proyecto será de gran utilidad en un futuro próximo para el desarrollo de la migración del sistema de comunicaciones de Actualize. Tras la presentación de esta memoria se proseguirá con pruebas piloto de validación del escenario seleccionado como primera opción (cloud).

VIII. REFERENCIAS

1) Amazon Web Services. *Simple Monthly Calculator* [en línea]. < http://calculator.s3.amazonaws.com/calc5.html> [Consulta: 12 de mayo de 2012]_____*Centro de Ahorro de AWS* [en línea]. < http://aws.amazon.com/es/economics/> [Consulta: 11 de mayo de 2012].

2) B. ALAN, Johnston. *SIP. Understanding the Session Initiation Protocol.* 3a ed. EEUU: Artech House, 2009. 395 p. ISBN: 978-1607839958

3) CARTER, G.; TS, J.; ECKSTEIN, R. *Using Samba: A File and Print Server for Linux, Unix & Mac OS X, 3rd Edition*. 3a ed. EEUU: O'Reilly Media, 2007. 448 p. ISBN: 978-0596007690

4) COLLINS, Daniel. *Carrier Grade Voice Over IP*. 2a ed. EEUU: The McGraw-Hill, 2002. 496 p. ISBN: 978-0071363266

5) Fortinet Inc. *FortiGate-60 Administration Guide*. [s.l.], 2004. 376 p.

6) ITU-CCITT. *Forcasting International Traffic* [en línea] Geneva, 1992. E.506 rev.1.

7) ITU-D SG 2/16 & ITC. *Teletraffic Engineering* [en línea]. 2001. Draft 2001-06-20

8) MADSEN, L.; VAN MEGGELEN, J.; RUSSELL, B. *Asterisk: The Definitive Guide*. EEUU: O'Reilly Media, 2011, 738 p. ISBN: 978-0596517342

9) NEMETH, E.; SNYDER, G.; HEIN, T. R.; (et al.). *UNIX and Linux System Administration Handbook.* . 4a ed. EEUU: Prentice Hall, 2010. 1344 p. ISBN: 978-0131480056

10) Rackspace Hosting. Rackspace Cloud Servers pricing. [en línea]. < http://www.rackspace.com/cloud/cloud_hosting_products/servers/pricing/> [Consulta: 10 de mayo de 2012].

11) ROBAR, Alex. *FreePBX 2.5 Powerful Telephony Solutions*. UK: Packt Publishing Ltd., 2009. 291 p. ISBN: 978-1-847194-72-5

12) Windows Azure. *Calculadora* [en línea]. < http://www.windowsazure.com/es-es/pricing/calculator/> [Consulta: 10 de mayo de 2012].

Referencias y fuentes

http://www.voip-info.org/
http://www.sinologic.net/blog/

Grupo de google Asterisk-ES:
https://groups.google.com/forum/?hl=es&fromgroups#!forum/asterisk-es

Forum de Asterisk y FreePBX:
http://forums.digium.com/
http://www.freepbx.org/forums

IX. ANEXOS

Apéndice A: Instalación de AsteriskNow e interfaz

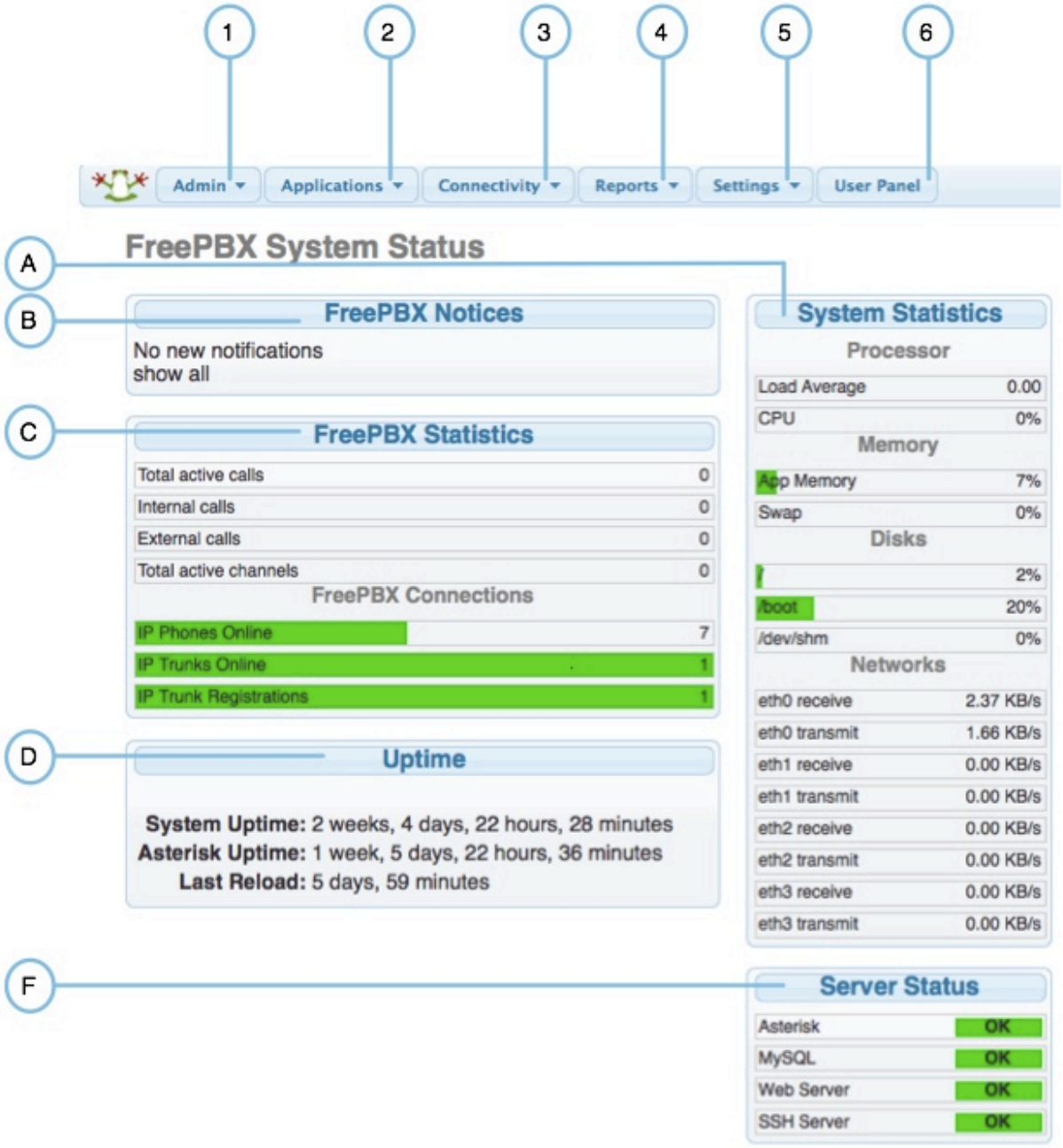
Para la instalación de AsteriskNow hemos descargado la imagen de disco de la página oficial de Asterisk. La primera pantalla del disco de instalación te da la opción de instalar sólo Asterisk o adicionalmente, instalar FreePBX. Esta última ha sido la opción elegida en nuestro caso.

Una vez la instalación es completada, tras autenticarnos como *root*, el siguiente paso es introducir en un navegador la IP del servidor para acceder a la interfaz gráfica que proporciona FreePBX.

Tras direccionar nuestro navegador a la IP anterior podemos comenzar con la configuración del sistema una vez nos autentifiquemos como administradores (admin/admin).

1. ***Admin***: Control y gestión de usuarios administrador, *back-up* y restauración, administración de módulos, grabaciones del sistema (buzones de voz, IVRs, etc.).
2. ***Applications***: Definición de comunicados, manejo de conjunto de DIDs (archivos .csv), manejo de conjunto de extensiones, habilitado y asignación de grabaciones, conferencias, DISA (Direct Inward System Access), creación y manejo de extensiones, FollowMe, colas, condiciones temporales, etc.
3. ***Conectivity***: configuración de *end-points* (teléfonos IP), creación y configuración de rutas de entrada y salida, trunks y..
4. ***Reports***: información sobre Asterisk (número de canales activos, *peers* online, etc.), *logs* de Asterisk, CDR (*Call Detail Record*), estado del sistema, impresión de extensiones.
5. ***Settings***: ajustes avanzados, SIP, IAX, generales, Asterisk API, *Music On Hold* (MOH), administración de buzón de voz, etc.

A. ***System Statistics***: datos de procesador, CPU, memoria consumida, capacidad en los discos, consumo en interfaces de red.
B. ***FreePBX Notices***: notificaciones relevantes como aparición de nuevas actualizaciones o mensajes de error o advertencia.
C. ***FreePBX Statistics***: número de llamadas activas (internas y externas), canales activos (SIP, IAX), disponibilidad de dispositivos, trunks.
D. ***Uptime*:** tiempos de trabajo.


1
2
3
4
5
6
Admin
Applications
Connectivity
Reports
Settings
User Panel
A
FreePBX System Status
B
FreePBX Notices
No new notifications
show all
C
FreePBX Statistics
Total active calls 0
Internal calls 0
External calls 0
Total active channels 0
FreePBX Connections
IP Phones Online 7
IP Trunks Online 1
IP Trunk Registrations 1
D
Uptime
System Uptime: 2 weeks, 4 days, 22 hours, 28 minutes
Asterisk Uptime: 1 week, 5 days, 22 hours, 36 minutes
Last Reload: 5 days, 59 minutes
System Statistics
Processor
Load Average 0.00
CPU 0%
Memory
App Memory 7%
Swap 0%
Disks
/ 2%
/boot 20%
/dev/shm 0%
Networks
eth0 receive 2.37 KB/s
eth0 transmit 1.66 KB/s
eth1 receive 0.00 KB/s
eth1 transmit 0.00 KB/s
eth2 receive 0.00 KB/s
eth2 transmit 0.00 KB/s
eth3 receive 0.00 KB/s
eth3 transmit 0.00 KB/s
F
Server Status
Asterisk OK
MySQL OK
Web Server OK
SSH Server OK

Apéndice B: tutorial X-lite

Objetivos del tutorial

A la hora de que dos o más extremos se comuniquen por voz utilizando como medio de transporte Internet (IP), es necesario definir los dispositivos con los que se llevará a cabo la comunicación. Existen varias soluciones, entre las que se encuentran los softphones. Un softphone no es más que un programa informático diseñado para ser usado desde un ordenador.

Existen multitud de compañías que ofrecen este tipo de programas. En nuestro caso haremos uso de X-lite, desarrollado por la empresa CounterPath Corporation.

Nuestro objetivo será realizar la instalación de este software en varios ordenadores y configurarlos de tal manera que puedan establecer comunicación entre ellos.

Nota: En caso de querer una instalación rápida recomiendo leer exclusivamente el punto 4: Proceso de Instalación.

¿Qué es X-Lite?

Se trata de un softphone desarrollado por la empresa canadiense CounterPath Corporation. Actualmente se encuentra en la versión 4.0 (2010). Entre las posibilidades que ofrece se encuentra la elección de los codecs a utilizar, ya sean de audio o video, siendo la cantidad de estos últimos mucho menor. En cuanto al protocolo de sesión que utiliza es SIP, no siendo posible elegir otro como pudiera ser IAX.

Desde el mismo panel principal podemos crear y editar contactos, ver un listado del registro de llamadas e incluso controlar el volumen de micrófono y altavoces mediante los iconos del panel superior.

Gráfico 12: X-Lite

Requerimientos

	Minimum	Optimal
Processor	Pentium 4® 2.4 GHz or equivalent	Intel Core Duo or equivalent, Video Card with DirectX 9.0c support
Memory	1 GB RAM	2 GB RAM
Hard Disk Space	50 MB	50 MB
Operating System*	Microsoft Windows XP Service Pack 2 Microsoft Windows Vista, 32-bits and 64-bits arch Microsoft Windows 7 Mac OS 10.5 or above	Microsoft Windows XP Service Pack 2 Microsoft Windows Vista, 32-bits and 64-bits arch Microsoft Windows 7 Mac OS 10.5 or above
Connection	IP network connection (broadband, LAN, wireless); Constant Internet connection	IP network connection (broadband, LAN, wireless); Constant Internet connection
Sound Adapter	Full-duplex, 16-bit or use USB Headset	Full-duplex, 16-bit or use USB Headset

Tabla 7: Requerimientos X-Lite (http://www.counterpath.com/x-lite.html)

Proceso de instalación

La instalación de X-lite es sencilla:

1. En primer lugar debemos descargarnos el software de la página de la compañía: http://www.counterpath.com/x-lite-download.html

2. Un vez tenemos descargado el programa lo ejecutaremos y seguiremos los pasos de instalación.

Configuración

En primer lugar debemos abrir la ventana de preferencias: pinchamos en Softphone>Account Settings

Una vez tenemos abierta la ventana de preferencias rellenamos los campos necesarios. Algunos de estos campos como pueda ser el de "proxy" tendrán valores diferentes según nos encontremos dentro de la misma Red Local (o conectados a través de VPN) o por el contrario estemos fuera de ella.

Configuración para usuarios DENTRO de la Red Local (o VPN):

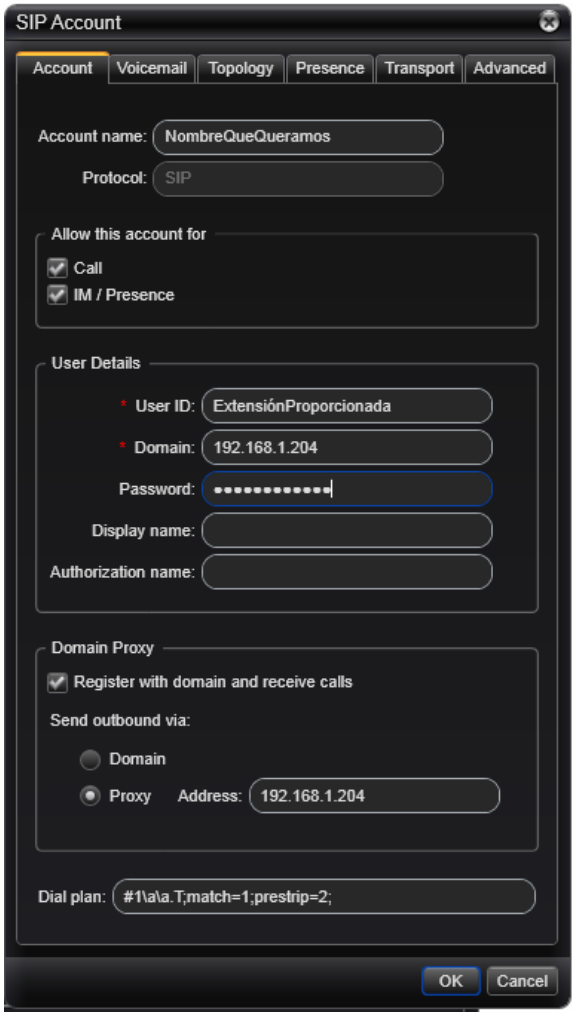


Gráfico 13: configuración X-Lite

Configuración para usuarios FUERA de la Red Local:

La IP correspondiente al campo *domain* y *proxy* debe ser nuestra IP pública.

Detalles de configuración:

- **Account name**: es irrelevante a la hora de autenticar con la centralita y podemos elegir el nombre de cuenta que nosotros creamos oportuno.
- **User ID**: Este campo es importante. El identificador que pongamos será el que Asterisk utilice para identificar quién llama aplicando la configuración y funcionalidades (redireccionamientos de llamada por ejemplo) especificadas en la configuración de Asterisk.
- **Password:** contraseña para autenticar el usuario en Asterisk.
- **Proxy**: dirección necesaria para establecer comunicación con Asterisk

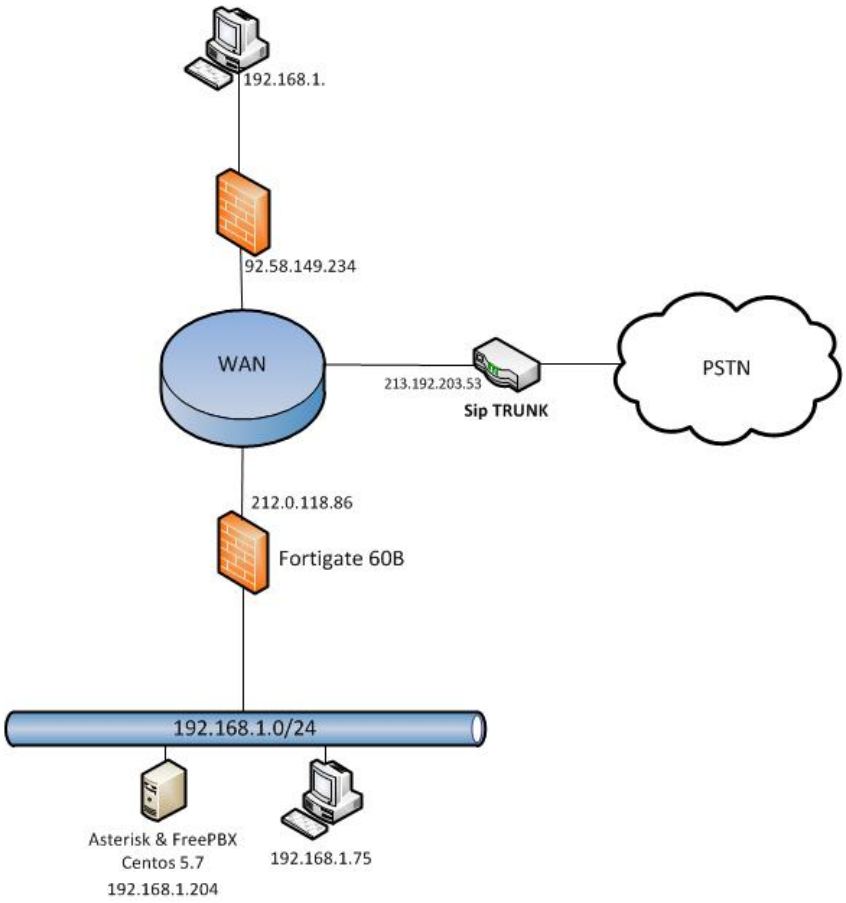
Apéndice C: Documentación capturas

Introducción

Tras la última prueba realizada (interfaz de red 3G en Asterisk) concluimos que el problema reside en el firewall de Actualize Fortigate60B. El tráfico de señalización transcurre correctamente mientras que el audio (RTP) falla en sentido wan1→internal. Es necesario hacer una captura detallada para ver el flujo de tráfico y así poder detectar la interfaz donde existen problemas.

En este documento puede encontrarse el diagrama de red del escenario en cuestión y una descripción detallada del proceso entero de resolución del problema.

Arquitectura de red



Proceso de captura

Para la traza del tráfico hemos utilizado las siguientes herramientas:

1. Wireshark
2. Sniffer de Fortigate
3. fgt2eth.exe

Sniffer de Fortigate

Tal como indica su nombre este programa propio de Fortigate muestra por salida de consola la traza del tráfico según las opciones que se hayan especificado a la hora de ejecutar el comando que lo ejecuta.

Sintaxis

```
diagnose sniffer packet <interfaz> '<filter>' <verbose> <num> a
```

- **interfaz**: podemos escoger entre las interfaces descritas para el Fortigate (internal o wan1)

- **filter**: [src/dst] host <host_name_o_IP> [udp/tcp/…[núm_de_puerto]]
  Estos parámetros son los básicos. Ellos pueden combinarse con el operador AND u OR (varios hosts).

- **verbose**: detalle que queremos de la traza. Puede escogerse del 1 al 6 siendo esta última la que más detalle aporta.

- **num**: el número de paquetes que **sniffer** lee antes de detenerse

- **a**: estampa de tiempo absoluto

fgt2eth.exe

Se trata de una aplicación propietaria de Fortigate. Con ella es posible convertir el archivo resultante de la captura con el sniffer de Fortigate a un archivo .pcap capaz de ser leído por Wireshark.

Debe ser ejecutado desde la consola de Windows mediante la siguiente sintaxis (previamente habrá que incluir fgt2eth.exe en el PATH de Windows y situarnos en el directorio donde se encuentren los archivos generados por el Sniffer de Fortigate):

- fgt2eth.exe –**i** *archivodeentrada* –**o** *archivodesalida*

donde *archivodeentrada* deberá tener el mismo nombre que el archivo generado por el Sniffer y *archivodeentrada* corresponde al nombre que queremos que tenga el .pcap a generar.

Procedimiento

Para la captura del tráfico ha sido necesario tener dos usuarios activos registrados en Asterisk que realicen llamadas en ambas direcciones. Los clientes SIP se encontraban (1) en la misma red local que el servidor Asterisk (SIP Client 1) y (2) en WAN (SIP Client 2) (tal como se describe en el **Gráfico 1**).

El tráfico ha sido capturado en las siguientes interfaces:

- 192.168.1.75 (SIP Client)
- 192.168.2.101 (SIP Client)
- wan1@Fortigate60B
- internal@Fortigate60B
- 192.168.1.204 (Servidor Asterisk)

Apéndice D: *pattern matching*

X: Representa cualquier número del 0 al 9

Z: Representa cualquier número del 1 al 9

N: Representa cualquier número del 2 al 9

[15-8]: corresponde a un solo número del rango de dígitos especificados. En este caso, el patrón correspondería a un solo 1, o a cualquier número entre el 5, 6, 7 y 8.

. (punto): representa uno o más caracteres sin importar cuales sean.

! (exclamación): representa cero o más caracteres sin importar cuales sean.